INTELLIGENT INSTRUMENTATION®

*INNOVATIVE PC Data Acquisition Solutions*

# UDAS-1001E Series Virtual Instrument (VI) Support Library User Manual for National Instruments® LabVIEW® Software

*A Burr-Brown Company*

**855M497 1.0**

Copyright 2000 by Intelligent Instrumentation Inc., Tucson, Arizona, USA

All rights reserved.

## Warranty and Repair Policy Statement

### General

Seller warrants that its products furnished hereunder will, at the time of delivery, be free from defects in material and workmanship and will conform to Seller's published specifications applicable at the time of sale. Seller's obligation or liability to Buyer for products which do not conform to above stated warranty shall be limited to Seller, at Seller's sole discretion, either repairing the product, replacing the product with a like or similar product, or refunding the purchase price of the nonconforming product, provided that written notice of said nonconformance is received by Seller within the time periods set forth below:

a. For all software products, including licensed programs, ninety (90) days from date of initial delivery to Buyer;

b. For all hardware products, including complete systems, one year from date of initial delivery to Buyer, subject to the additional conditions of paragraphs c) and d) below;

c. In the event that Buyer's returned product is a Discontinued product and is irreparable for any reason, Seller may elect to replace it with like or similar product that is, in Seller's sole judgment, the closest equivalent to the returned product. Seller does not warrant that such replacement product will be an exact functional replacement of the returned product.

Further, all products warranted hereunder for which Seller has received timely notice of nonconformance must be returned FOB Seller's plant no later than thirty (30) days after the expiration of the warranty periods set forth above.

These warranties provided herein shall not apply to any products which Seller determines have been subjected, by Buyer or others, to operating and/or environmental conditions in excess of the limits established in Seller's published specifications or otherwise have been the subject of mishandling, misuse, neglect, improper testing, repair, alteration or damage. THESE WARRANTIES EXTEND TO BUYER ONLY AND NOT TO BUYER'S CUSTOMERS OR USERS OF BUYER'S PRODUCT AND ARE IN LIEU OF ALL OTHER WARRANTIES WHETHER EXPRESS, IMPLIED OR STATUTORY INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SELLER BE LIABLE FOR INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES. Seller's liability for any claim of any kind shall in no case exceed the obligation or liability specified in this Warranty clause.

## Technical Assistance and Service

Seller's warranty as herein set forth shall not be enlarged, diminished or affected by, and no obligation or liability shall arise or grow out of, Seller's rendering of technical advice, facilities or service in connection with Buyer's order of the goods furnished hereunder. Products returned for warranty service, but which are found to be fully functional and in conformance with specifications may be subject to a nominal service charge and return freight charges. Periodic re-calibration of products, if required, is the responsibility of Buyer and is not provided under this Warranty.

### Online Support

If errors persist after repeatedly checking the unit as described in the *UDAS-1001E Series Hardware User Manual*, check the technical support links on Intelligent Instrumentation's 24-hour World Wide Web site at **http:// www.instrument.com**. The site contains information on Intelligent Instrumentation's products, new developments, announcements, application notes, application examples, and other useful information. The site and support areas continue to grow as new products, updates, and features are added.

### Email Support

Intelligent Instrumentation's technical support can be reached via email. When sending an email message, be sure to include complete contact information as well as a detailed description of the products being used and the problem. Use the following addresses to contact the support group:

**North American Customers**

us_support@instrument.com

app_eng@instrument.com

**International Customers**

international@instrument.com

int_support@instrument.com

In France:france@instrument.com

In Italy: italy@instrument.com

In Germany: germany@instrument.com

## Static Sensitivity

Seller ships all static-susceptible products in anti-static packages. Seller's Warranty as herein set forth shall not cover warranty repair or replacement for products damaged by static due to Buyer's failure to use proper protective procedures when handling, storing, or installing products.

## Trademarks

U-Link ™Software Libraries is a trademark of Intelligent Instrumentation Incorporated.

VisualDesigner® is a registered trademark of Intelligent Instrumentation Incorporated.

LabVIEW® is a registered trademark of National Instruments Corporation.

Windows® 98 is a registered trademark of Microsoft® Corporation in the U.S.A. and other countries.

Other products or brand names are trademarks or registered trademarks of their respective companies.

## Use of Equipment

Intelligent Instrumentation Inc., assumes no responsibility for any direct, indirect or consequential loss or damages resulting from misuse of the equipment or for improper or inadequate maintenance of the equipment or for any such damage or loss resulting from the use of other equipment, attachments, accessories, and repairs at any time made to or placed upon the equipment or any replacement thereof. Furthermore, Intelligent Instrumentation Inc., makes no representations or warranties, either expressed or implied, in connection with the use of the equipment in the event it is improperly used, repaired or maintained.

## FCC Radio Frequency Interference Statement

This equipment generates and uses radio frequency energy, and may cause interference to radio or television reception.

Per FCC rules, Part 15, Subpart J, operation of this equipment is subject to the conditions that no harmful interference is caused and that interference must be accepted that may be caused by other incidental or restricted radiation devices, industrial, scientific or medical equipment, or from any authorized radio user.

The operator of a computing device may be required to stop operating his device upon a finding that the device is causing harmful interference and it is in the public interest to stop operation until the interference problem has been corrected.

The user of this equipment is responsible for any interference to radio or television reception caused by the equipment. It is the responsibility of the user to correct such interference.

**UDAS-1001E Series User Manual**
**Revision History**

| Version | Date | Revision |
|---|---|---|
| 1.0 | 000301 | Original Release |

# Table of Contents


**Chapter 1 Introduction**

1.1 About this Manual . . . . . 1-1
1.2 LabVIEW UDAS Support Library . . . . . 1-2
1.3 Online Help . . . . . 1-2

**Chapter 2 Support Library Installation and Files**

2.1 Introduction . . . . . 2-1
2.2 System Configuration Requirements . . . . . 2-1
2.3 The Software License Agreement . . . . . 2-1
2.4 Software Installation . . . . . 2-2
2.4.1 Installation Procedures . . . . . 2-2
2.5 Function Block and VI File Listing . . . . . 2-2
2.5.1 LabVIEW Directory . . . . . 2-3
2.5.2 UDAS Support Library Directory . . . . . 2-3
2.5.3 Sample Applications Directory . . . . . 2-6
2.5.4 Function VI Icons . . . . . 2-8

**Chapter 3 UDAS Support Library: Mandatory & Optional Functions**

3.1 Introduction . . . . . 3-1
3.2 Function Wiring Connection Information . . . . . 3-1
3.3 Mandatory Functions . . . . . 3-2
3.3.1 Initialization Functions . . . . . 3-2
3.3.1.1 uDeviceOpen . . . . . 3-2
FIGURE 3.1 uDeviceOpen function . . . . . 3-2
3.3.1.2 uDeviceClose . . . . . 3-3
FIGURE 3.2 uDeviceClose function . . . . . 3-3
3.4 Optional Functions . . . . . 3-3
3.4.1 uSYSInquire . . . . . 3-4
FIGURE 3.3 uSYSInquire function . . . . . 3-4
3.5 Function Sequence . . . . . 3-4
3.5.1 Function Execution Sequencing . . . . . 3-5
3.6 Sample Program . . . . . 3-5
FIGURE 3.4 Diagram of a Sample-Based Program . . . . . 3-6
FIGURE 3.5 Front Panel Snapshot from a Driver-Based Sample Program 3-7
3.6.1 Error Code Input and Error Code Output Terminals . . . . . 3-7
FIGURE 3.6 Error Code Boolean Case Structure (True) . . . . . 3-8
FIGURE 3.7 Error Code Boolean Case Structure (False) . . . . . 3-9
3.7 Copying Examples as the Basis of Applications . . . . . 3-10
FIGURE 3.8 UDAS Function Sample Files . . . . . 3-10
3.8 UDAS Support Library Functions . . . . . 3-12
FIGURE 3.9 LabVIEW Functions Palette . . . . . 3-12

FIGURE 3.10 User Libraries Palette . . . . . 3-13
FIGURE 3.11 UDAS Support Palette . . . . . 3-13

**Chapter 4 Support Library: Analog Input & High-Speed Analog Input Functions**
4.1 Analog Input Functions . . . . . 4-1
4.1.1 uAIConfigureList . . . . . 4-3
FIGURE 4.1 uAIConfigureList Function . . . . . 4-3
4.1.2 uAIRead . . . . . 4-4
FIGURE 4.2 uAIRead Function . . . . . 4-4
4.1.3 uAIMAXEnable . . . . . 4-5
FIGURE 4.3 uAIMAXEnable Function . . . . . 4-5
4.2 High-Speed Analog Input Functions . . . . . 4-6
FIGURE 4.4 uFrequencyToRGCounts Function . . . . . 4-8
FIGURE 4.5 uRGConfigure Function . . . . . 4-8
FIGURE 4.6 uRGEnable Function . . . . . 4-9
4.2.1 uAIHSConfigureList . . . . . 4-10
FIGURE 4.7 uAIHSConfigureList Function . . . . . 4-10
4.2.2 uAIHSEnable . . . . . 4-13
FIGURE 4.8 uAIHSEnable Function . . . . . 4-13
4.2.3 uAIHSReturn . . . . . 4-13
FIGURE 4.9 uAIHSReturn Function . . . . . 4-13
4.2.4 uAIHSRead . . . . . 4-14
FIGURE 4.10 uAIHSRead Function . . . . . 4-14
4.2.5 uAIHSStatus . . . . . 4-15
FIGURE 4.11 uAIHSStatus Function . . . . . 4-15

**Chapter 5 Support Library: Analog Output & High-Speed AO Functions**
5.1 Analog Output Functions . . . . . 5-1
5.1.1 uAOConfigureList . . . . . 5-1
FIGURE 5.1 uAOConfigureList Function . . . . . 5-1
5.1.2 uAOWrite . . . . . 5-2
FIGURE 5.2 uAOWrite Function . . . . . 5-2
5.2 High-Speed Analog Output Functions . . . . . 5-3
FIGURE 5.3 uRGConfigure Function . . . . . 5-5
5.2.1 uAOHSConfigureList . . . . . 5-6
FIGURE 5.4 uAOHSConfigureList Function . . . . . 5-6
5.2.2 uAOHSEnable . . . . . 5-6
FIGURE 5.5 uAOHSEnable Function . . . . . 5-6
5.2.3 uAOHSStatus . . . . . 5-7
FIGURE 5.6 uAOHSStatus Function . . . . . 5-7
5.2.4 uAOHSWrite . . . . . 5-7

FIGURE 5.7 uAOHSWrite Function . . . . . 5-7

**Chapter 6 Support Library: Digital I/O Port & Bit Functions**

6.1 Introduction . . . . . 6-1
6.2 Digital Input/Output Port Functions . . . . . 6-1
6.2.1 uDIOConfigureList . . . . . 6-1
FIGURE 6.1 uDIOConfigureList Function . . . . . 6-2
6.2.2 uDIORead . . . . . 6-2
FIGURE 6.2 uDIORead Function . . . . . 6-2
6.2.3 uDIOWrite . . . . . 6-3
FIGURE 6.3 uDIOWrite Function . . . . . 6-3
6.3 Digital Input/Output Bit Functions . . . . . 6-4
6.3.1 uDIOBITConfigureList . . . . . 6-4
FIGURE 6.4 uDIOBITConfigureList Function . . . . . 6-4
6.3.2 uDIOBITRead . . . . . 6-6
FIGURE 6.5 uDIOBITRead Function . . . . . 6-6
6.3.3 uDIOBITWrite . . . . . 6-7
FIGURE 6.6 uDIOBITWrite Function . . . . . 6-7

**Chapter 7 Support Library: Hardware Counter Functions**

7.1 Introduction . . . . . 7-1
7.2 uCTRConfigure . . . . . 7-2
FIGURE 7.1 uCTRConfigure Function . . . . . 7-2
7.3 uCTREnable . . . . . 7-3
FIGURE 7.2 uCTREnable Function . . . . . 7-3
7.4 uCTRRead . . . . . 7-4
FIGURE 7.3 uCTRRead Function . . . . . 7-4
7.5 uCTRStatus . . . . . 7-5
FIGURE 7.4 uCTRStatus Function . . . . . 7-5

**Chapter 8 Support Library: Thermocouple Functions**

8.1 Thermocouple Functions . . . . . 8-1
8.1.1 uCountsToTemperature . . . . . 8-2
FIGURE 8.1 uCountsToTemperature Function . . . . . 8-2
8.1.2 uCountsToTemperatureA . . . . . 8-2
FIGURE 8.2 uCountsToTemperatureA Function . . . . . 8-2

**Appendix A Error Codes**

A.1 Introduction . . . . . A-1
A.1.1 Explanation of Error Code Values . . . . . A-1
A.2 List of Error Codes . . . . . A-1

# Chapter 1 Introduction

## 1.1 About this Manual

This manual is a guide to using the UDAS Support Library for National Instruments LabVIEW®, a Virtual Instrument (VI) library for Intelligent Instrumentation's UDAS-1001E family of Data Acquisition Systems. This information is organized into chapters describing how to install and use the LabVIEW UDAS Support Library. These VI functions are required to create UDAS data acquisition and control applications under Windows 98, using National Instruments LabVIEW 5.0 or higher software.

The contents of this manual includes:

Chapter 1 Provides an introduction to this manual and a brief overview.

Chapter 2 Includes instructions for installing the UDAS Support Library for LabVIEW.

Chapter 3 Describes how to create LabVIEW applications using the UDAS Support Library low-level VIs.

Chapter 4 Contains information on how to use the analog input functions, and high-speed analog input functions.

Chapter 5 Includes information on how to use the analog output functions, and high-speed analog output functions.

Chpater 6 Contains information on how to use the digital input/output port functions and digital input/output bit functions.

Chapter 7 Includes information on how to use the hardware counter functions.

Chapter 8 Contains information on how to use the thermocouple functions.

Appendix A Contains error code information.

**Note:** To use this manual, a prior knowledge of LabVIEW concepts and terminology is required. Some familiarity with the general features and functions of UDAS -1001E and the U-Link Support Libraries is helpful.

## 1.2 LabVIEW UDAS Support Library

The UDAS Support Library consists of a set of low-level LabVIEW VIs which correspond to I/O functions available on a UDAS unit.

> **Note:** Review the README.TXT file supplied with the software; it may contain new information not included in this manual.

Each low-level VI in the UDAS Support Library for LabVIEW (UDAS-1001E) corresponds to a UDAS U-Link Software Libraries (UDAS -1001E) driver function call. Every LabVIEW UDAS Support VI generates a 32-bit integer result, or error code, indicating the success or failure of the function. VIs contain error code inputs and outputs for passing error information within the program and for regulating execution flow.

If you are familiar with the U-Link driver and LabVIEW, creating applications with the Support Library VI, is easier and more efficient than using the traditional coding techniques required when using the driver. Sample application VI's that demonstrate common applications are included with the Support Library. These examples are very useful as templates for custom applications.

The UDAS Support Library includes function calls for:

- Configuring a UDAS unit to perform a particular type of data acquisition.
- Calls for reading and writing analog and digital data.
- Various utility functions for simplifying routine operations.

## 1.3 Online Help

Online help is available for each function in the UDAS Support Library. The easiest way to access the online help in LabVIEW is when the desired function is selected, copied, and inserted into the diagram. Right-click on the function icon and the help page for that function opens.

Online help can also be accessed through the Windows Start menu and can be accessed by clicking Start\Programs\U-Link 1.0 LabVIEW Support. Click the UDAS Support Online Manual shortcut and the *Help Topics: Help for LabVIEW UDAS Support Library* dialog opens. The *Contents* tab is the active dialog displayed which contains a list of items beginning with the contents, followed by the different types of functions, and the error codes. To open any of these items on this list, either select the item and then click on the *Display* button, or double-click the item.

The *Contents* page contains a brief overview of the Support Library, all of the function icons, as they appear in the Functions Palette and a list of the types of functions that link to the help pages for the actual functions. Click on the desired name of one of the individual functions to open the help page for that function. If you are familiar with the function icons, click the icon and the help page for that function opens.

The *Error Codes* page contains numbered error codes returned by the UDAS lower level VIs and their descriptions. Information on error codes is provided in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions* and Appendix A, *Error Codes*.

The information on the functions and error codes can be printed, if desired.

By default, the online help file is located on the local drive:\Program Files\National Instruments\LabVIEW\User.lib\UDAS Support\u-link.HLP.

# Chapter 2 Support Library Installation and Files

## 2.1 Introduction

The LabVIEW UDAS Support Library and related files, such as the on-line help and sample program files are compressed on the supplied UDAS-1001E Data Acquisition System CD.

This chapter lists the following:

- System configuration requirements
- Program license agreement
- Provides installation instructions
- Lists the names of the installed files

## 2.2 System Configuration Requirements

The minimum system requirements include:

- Personal computer 486-based or newer system using Windows 98 with a minimum of 8 Mbytes of RAM (16 Mbytes or more is recommended).
- Install LabVIEW Version 5.0 on the system prior to installing the LabVIEW UDAS Support Library. The LabVIEW UDAS Support Library installation requires approximately 3 Mbytes of free hard disk space.

## 2.3 The Software License Agreement

Read the PROGRAM LICENSE AGREEMENT printed on the envelope containing the UDAS-1001E Data Acquisition System CD ROM. The license agreement is also located in the Root Directory on the CD. If you do not understand the license agreement or have any questions, contact your sales representative before continuing.

# 2.4 Software Installation

## 2.4.1 Installation Procedures

The UDAS-1001E Data Acquisition System CD contains the Support Library and related files.

To install the software onto the hard drive:

1. Close all Windows applications that are running.
2. Insert the UDAS-1001E Data Acquisition System CD in the CD ROM drive on the computer.
3. Select Start\Run and in the Run dialog, enter the drive letter for the CDROM:

   **LabVIEW\SETUP.EXE** and click the *OK* button to start the setup program. Follow the installation instructions.

> **Note:** To directly access the UDAS VIs from the LabVIEW Function Palette, the UDAS support library must be installed in the correct folder.

4. When installation is complete, a program group named U-Link LabVIEW Support (by default) is created, containing shortcuts to the following:
   - UDAS Support Online Manual - UDAS VI Reference in a Windows Help format.
   - Readme File - Contains latest information on the software and tech support telephone numbers.
   - Uninstall U-Link 1.0 LabVIEW Support - Use this program to uninstall UDAS Support.

# 2.5 Function Block and VI File Listing

This section briefly describes the support library files which include the following three directories:

- Existing LabVIEW Directory
- UDAS Support Library Directory
- Sample Applications Directory

Each directory contains the name and description of the files included in that directory.

## 2.5.1 LabVIEW Directory

The LabVIEW directory contains U-Link driver DLL and Readme file. The default directory follows the path: **C:\Program Files\National Instruments\Labview**

The following files are included in the LabVIEW directory: I3USB.DLL and UDAS README.TXT.

#### I3USB.DLL

This is a U-Link driver that implements all of the UDAS-related functions. This file should always be installed in the LabVIEW directory.

#### UDAS README.TXT

This file contains important information about the LabVIEW UDAS Support Software Library. Examine the *UDAS Readme.txt* file for the latest information. To access this file using the shortcut path, in the Windows Start menu click Start\Programs\U-Link 1.0 LabVIEW Support. Click the Readme file shortcut and the Readme text file opens.

The *Contents* tab is the active dialog displayed which contains a list of items beginning with the contents, followed by the different types of functions, and the error codes. To open any of these items on this list, either select the item and then click on the *Display* button, or double-click the item. This file may be read through any ASCII text editing program.

## 2.5.2 UDAS Support Library Directory

The UDAS Support Library Directory contains the actual UDAS support low-level VIs and online help (**u-link.HLP** and **u-link.CNT**) files. To access the UDAS Support Library Directory, follow the path: **Labview\User.lib\UDAS Support**

#### UDAS Support Online Reference Manuals

The UDAS Support Online reference manual help files include the **u-link.HLP** and **u-link.CNT** files. Descriptions of each function and error code description are included in the u-link.HLP file.

## UDAS Support Functions

The following is an alphabetical listing of the UDAS Support Functions.

| Function | Description |
|---|---|
| **uAIConfigureList.vi** | Configures the gain, range, and so forth, in a list of analog input channels. |
| **uAIHSConfigureList.vi** | Configures the gain, range, and so forth, in a list of analog input channels for high-speed analog input acquisition. |
| **uAIHSEnable.vi** | Enables or disables a high-speed analog input process, configures number of samples. |
| **uAIHSRead.vi** | Reads the data acquired in a high-speed analog input process. |
| **uAIHSReturn.vi** | Returns data acquired in a high-speed analog input process from device buffers. |
| **uAIHSStatus.vi** | Obtains status and buffer information on a high-speed analog input process. |
| **uAIMAXEnable.vi** | Enables the specified termination panels on the AI-MAX 256 channel analog input expansion system. |
| **uAIRead.vi** | Reads a previously configured analog input channel on specified termination panel (in non-high-speed mode) or all panels, or all analog input channels. |
| **uAOConfigureList.vi** | Configures (channel, range) a list of analog output channels. |
| **uAOHSConfigureList.vi** | Configures a high-speed analog output process. |
| **uAOHSEnable.vi** | Enables or disables a high-speed analog output process and transmits the data to the output. |
| uAOHSStatus.vi | Obtains the status information for a high-speed analog output process. |
| **uAOWrite.vi** | Writes data to a previously configured analog output channel or to all channels. |
| **uAOHSWrite.vi** | Downloads a segment of a continued analog output waveform to the device. |
| **uCountsToTemperature.vi** | Converts a thermocouple and cold-junction counts data value to a temperature value. |
| **uCountsToTemperatureA.vi** | Converts an array of thermocouple and cold-junction counts data values to an array of temperature values. |
| **uCountsToVolts.vi** | Converts an analog input counts data value to a floating point voltage value. |

| Function | Description |
|---|---|
| **uCountsToVoltsA.vi** | Converts the array of the specified counts data values to an array of floating point voltage values. |
| **uCTRConfigure.vi** | Used to configure a 16-bit hardware counter on a UDAS unit for a specified counter mode. |
| **uCTREnable.vi** | Starts or stops an event counter, or disables a frequency or time measurement. |
| **uCTRRead.vi** | Reads the current count value, starts the measurement, or returns the result from a complete measurement. |
| **uCTRStatus.vi** | Returns the current status of the event-counting, or measurement process. |
| **uDeviceClose.vi** | De-initializes the UDAS device. |
| **uDeviceOpen.vi** | Initializes the UDAS device. |
| **uDIOBITConfigureList.vi** | Configures a list of digital I/O port bits. |
| **uDIOBITRead.vi** | Reads a digital input port bit configured for any mode. |
| **uDIOBITWrite.vi** | Writes data to a digital output port bit. |
| **uDIOConfigureList.vi** | Configures (input or output) a list of digital I/O ports. |
| **uDIORead.vi** | Reads one or two digital I/O ports configured for input. |
| **uDIOWrite.vi** | Writes data to one or two digital output ports. |
| **uFrequencyToRGCounts.vi** | Computes the count values used to program a rate generator for a specified output frequency. |
| **uRGConfigure.vi** | Configures the analog input or analog output rate generator channel with an initial count value. |
| **uRGEnable.vi** | Enables or disables a rate generator channel. |
| **uSetOption.vi** | Sets optional parameters on a UDAS unit. |
| **uSYSInquire.vi** | Obtains information on the UDAS unit. |
| **uTRIGConfigure.vi** | Configures an analog input channel as a trigger for a high-speed analog input or output process on a UDAS unit. |
| **uVoltsToCounts.vi** | Converts a floating point voltage value to counts data value (for use by the analog output write and trigger configuration functions). |
| **uVoltsToCountsA.vi** | Converts the specified array of floating point voltage values to an array of counts data value (for use by the analog output write and trigger configuration functions). |

## 2.5.3 Sample Applications Directory

The Sample Applications Directory contains examples of analog input, analog output, digital input bit-level, digital output bit-level, and digital I/O port-level, high-speed analog input, thermocouple temperature measurement, high-speed analog output, hardware counter, and continuous high-speed analog input. To access the Sample Applications Directory, follow the path: **C:\Program Files\National Instruments\Labview\UDAS Examples**

### Labview\UDAS Examples

#### UDAS Analog Input Sample.vi

This sample VI demonstrates how to acquire data from multiple analog input channels on a UDAS unit. The sample features the **uAIConfigureList.vi**, **uAIRead.vi** and the **uCountsToVoltsA.vi**.

#### UDAS Analog Output Sample.vi

This sample VI demonstrates how to write data to analog output channels on a UDAS unit and features the **uAOConfigureList.vi** and **uAOWrite.vi** and the **uVoltsToCountsA.vi**.

#### UDAS Continuous HS Analog Input Sample.vi

This sample VI demonstrates how to perform continuous high-speed acquisition of analog data on a UDAS unit. To terminate the data acquisition process at any time, press the *Stop* button. This sample features the **uAIHSConfigureList.vi**, **uAIHSEnable.vi**, **uAIHSReturn.vi**, **uAIHSRead.vi**, **uRGConfigure.vi**, **uRGEnable.vi**, **uFrequencyToRGCounts.vi**, and **uCountsToVoltsA.vi**.

#### UDAS Digital Input Bit Sample.vi

This sample VI demonstrates how to use digital input port bits on an UDAS unit. Select either the Digital Input mode or one of the trigger modes. Uses **uDIOBitWrite.vi** to arm the triggers. The sample features the **uDIOBITConfigureList.vi** and **uDIOBITRead.vi** UDAS support functions.

#### UDAS Digital IO Sample.vi

This sample VI demonstrates how to read or write digital data either from or to ports on a UDAS unit and features the **uDIOConfigureList.vi**, **uDIORead.vi** and **uDIOWrite.vi** UDAS support functions.

#### UDAS Digital Output Bit Sample.vi

This sample VI demonstrates how to use digital output port bits on a UDAS unit and features the **uDIOBITConfigureList.vi**, and **uDIOBITWrite.vi** UDAS support functions.

### Hardware Counter Sample.vi

This sample VI demonstrates how to use a hardware counter on a UDAS unit. This sample features **uCTRConfigure.vi**, **uCTREnable.vi**, **uCTRRead.vi**.

### UDAS High-Speed Analog Input Sample.vi

This sample VI demonstrates how to perform high-speed acquisition of analog data on a UDAS unit until a specified number channel is acquired. The sample features the **uAIHSConfigureList.vi**, **uAIHSEnable.vi**, **uAIHSRead.vi**, **uAIHSReturn.vi**, **uRGConfigure.vi**, **uRGEnable.vi**, **uFrequencyToRGCounts.vi**, and **uCountsToVoltsA.vi**.

### High-Speed Analog Output Sample.vi

This sample features the **uAOHSConfigureList.vi**, **uAOHSEnable.vi**, **uAOHSStatus.vi** **uFrequencyToRGCounts.vi**, **uRGConfigure.vi**, and **uCountsToVoltsA.vi** utilities. This sample continuously writes sine wave, triangle wave-forms to UDAS analog outputs. **Note:** LabVIEW Advanced Analysis library should be installed as part of LabVIEW.

### UDAS Thermocouple Sample.vi

This sample VI demonstrates how to obtain thermocouple temperature measurements using a UDAS unit. It features the **uAIConfigureList.vi** and **uAIRead.vi** UDAS support functions and the **uCountsToTemperatureA.vi** utility.

**Typej.tc** Conversion table for standard Type J thermocouples.

**Typej5b.tc** Conversion table for PCI-5B37J non-linearized thermocouple signal conditioning block.

**Typek.tc** Conversion table for standard Type K thermocouples.

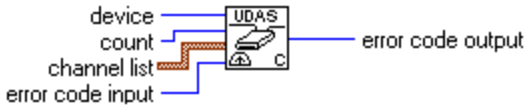
**Typek5b.tc** Conversion table for PCI-5B37K non-linearized thermocouple signal conditioning block.

**Typet.tc** Conversion table for standard Type T thermocouples.

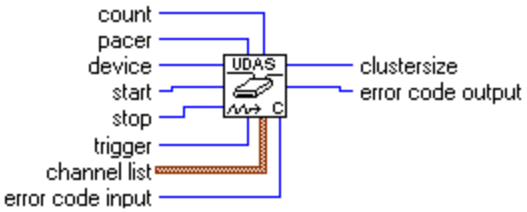
**Typet5b.tc** Conversion table for PCI-5B37T non-linearized thermocouple signal conditioning block.

## 2.5.4 Function VI Icons

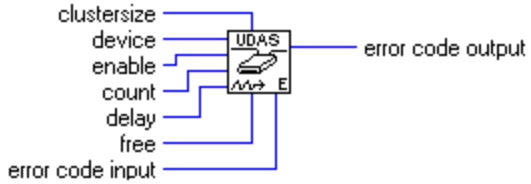
### uAIConfigureList.vi



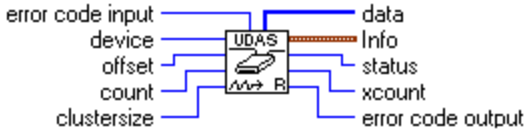

### uAIHSConfigureList.vi



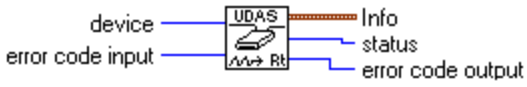

### uAIHSEnable.vi



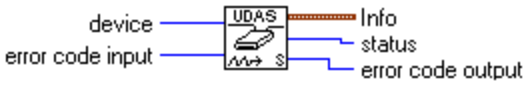

### uAIHSRead.vi



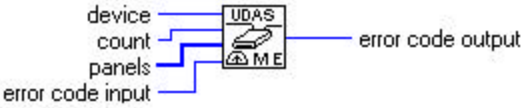

### uAIHSReturn.vi



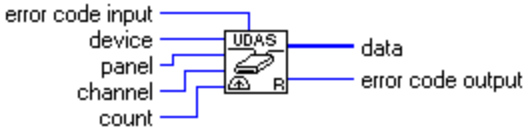
### uAIHSStatus.vi



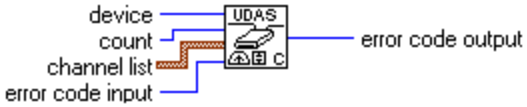

### uAIMAXEnable.vi



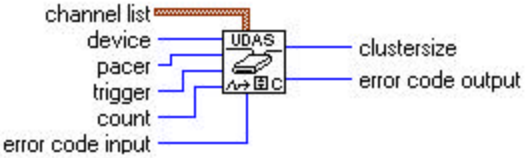

### uAIRead.vi



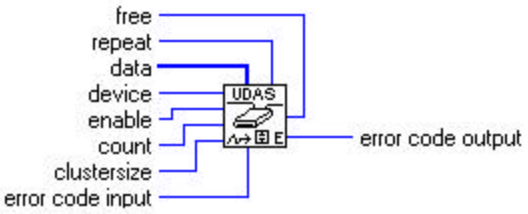

### uAOConfigureList.vi




### uAOHSConfigureList.vi

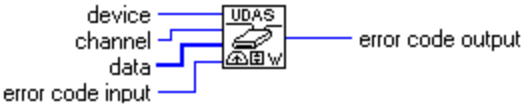


### uAOHSEnable.vi

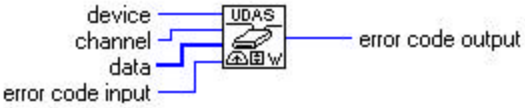



### uAOHSStatus.vi

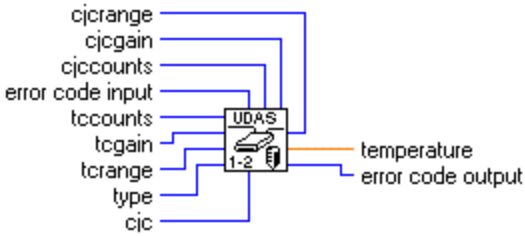
### uAOWrite.vi



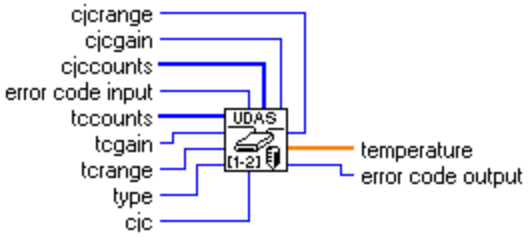

### uAOHSWrite.vi



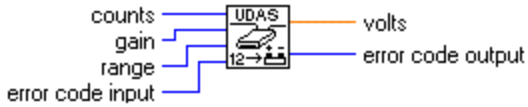

### uCountsToTemperature.vi



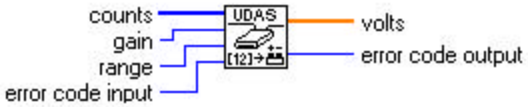
### uCountsToTemperatureA.vi



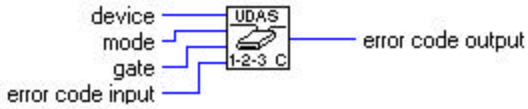

### uCountsToVolts.vi




### uCountsToVoltsA.vi



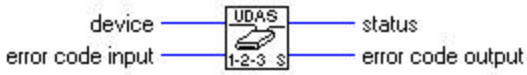

### uCTRConfigure.vi




### uCTREnable.vi

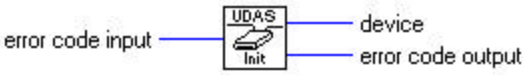
### uCTRRead.vi

### uCTRStatus.vi

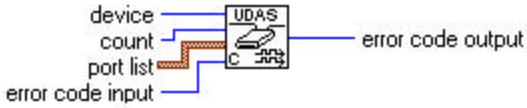



### uDeviceClose.vi

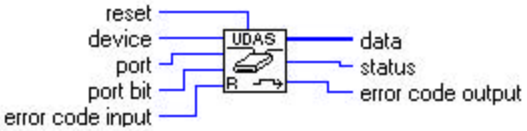
### uDeviceOpen.vi



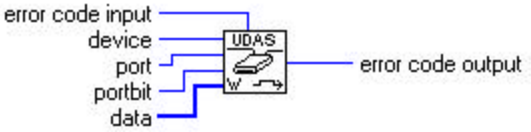

### uDIOBITConfigureList.vi



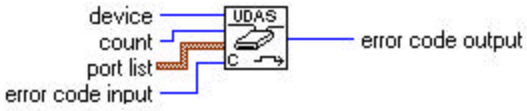

### uDIOBITRead.vi



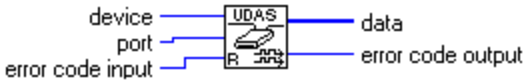

### uDIOBITWrite.vi



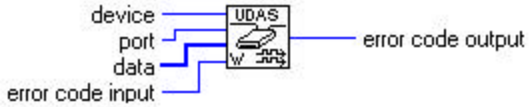
### uDIOConfigureList.vi



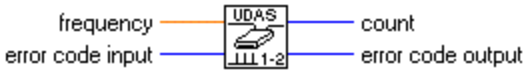

### uDIORead.vi



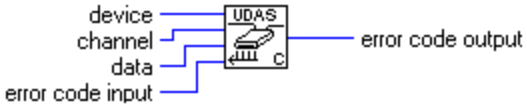

### uDIOWrite.vi



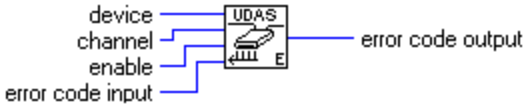

### uFrequencyToRGCounts.vi



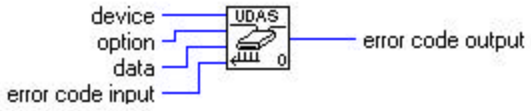

### uRGConfigure.vi




### uRGEnable.vi

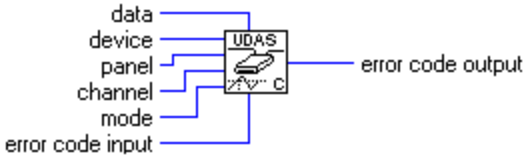


### uSetOption.vi




### uSYSInquire.vi

### uTRIGConfigure.vi




### uVoltsToCounts.vi

### uVoltsToCountsA.vi

# Chapter 3 UDAS Support Library: Mandatory & Optional Functions

## 3.1 Introduction

The Support Library consists of 38 function VIs for the UDAS-1001E Series Data Acquisition Systems. The functions are listed according to their type, and detailed information on each function is located in the U-Link 1.0 LabVIEW Support program help topics.

The functions in the UDAS library are used to program I/O tasks, such as analog input/output and digital input/output. UDAS Support functions have two common terminals; error code input, and error code output. These terminals specify program flow, reporting and handling errors.

Creating an application with the LabVIEW UDAS Support Library consists of inserting and wiring the desired function VIs into the diagram and running the front panel application. This chapter provides information on the mandatory functions that are used in every diagram, and explains the sequence of steps necessary to perform a desired data acquisition process.

## 3.2 Function Wiring Connection Information

Descriptions for each of the input and output items are provided in the online help page for this function. The easiest way to access the online help in LabVIEW is when the desired function is selected, copied, and inserted into the diagram. Right-click on the function icon and the help page for that function opens.

Online help can also be accessed by clicking Start\Programs\U-Link 1.0 LabVIEW Support\UDAS Support Online Manual. Click the UDAS Support Online Manual and the *Help Topics: Help for LabVIEW UDAS Support Library* dialog opens. The *Contents* tab is the active dialog displayed which contains a list of items beginning with the contents, followed by the different types of functions, and the error codes. To open any of these items on this list, either select the item and then click on the *Display* button, or double-click the item.

# 3.3 Mandatory Functions

## 3.3.1 Initialization Functions

There are two mandatory initialization functions, **uDeviceOpen** and **uDeviceClose**, that must be included in every application diagram.

### 3.3.1.1 uDeviceOpen

This is always the first function (shown in FIGURE 3.1, *uDeviceOpen function*) that is included in every diagram. This function opens the UDAS device and performs an initial test. The error code enters the function through the input, and must pass through the output to the next function in the diagram. The UDAS device is initialized by this function. The error code does not enter this function unless multiple UDAS units are being used simultaneously.

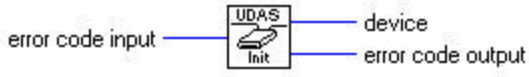



**FIGURE 3.1 uDeviceOpen function**

#### Accessing the Help Page for this Function

Access the online Help manual as described in section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Initialization Functions* item and select **uDeviceOpen**.

> **Note:** If more than one UDAS unit is connected to the host computer, multiple calls to the uDeviceOpen.vi may be required to open a particular UDAS unit.

### 3.3.1.2 uDeviceClose

This is always the last function (FIGURE 3.2, *uDeviceClose function*) included in every diagram. It closes the handle to the UDAS device and the error code is not passed beyond this function unless multiple UDAS units are being used simultaneously.

**FIGURE 3.2 uDeviceClose function**

After completing the I/O functions (such as uAIHSConfigureList, uAIHSEnable, and so forth), for the UDAS unit, wire the **uDeviceClose** function in the diagram to de-initialize the UDAS device.

#### Accessing the Help Page for this Function

Access the online Help manual as described in section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Initialization Functions* item and select **uDeviceClose**.

## 3.4 Optional Functions

There are two optional functions, **uSYSInquire** and **uSetOption**, that provide information on each UDAS unit and sets optional parameters on a UDAS unit, respectively.

The **uSYSInquire** function is an optional function that returns information about the connected UDAS unit. This function is used to obtain the UDAS board identification number, serial number, hardware and firmware versions, and detailed information on UDAS internal errors. Although this is an optional function, including it in each diagram is recommended.

In a typical diagram, the **uSYSInquire** function follows the **uDeviceOpen** function. Refer to the UDAS **Analog Input Sample.vi** Diagram to view the use of these functions in a diagram.

The **uSetOption** is another optional function. It sets optional parameters on a UDAS unit. The UDAS unit has two rate generators; one is for high-speed data acquisition, and the other is for high-speed analog output. This option enables the selection of either rate generator for the termination panel. Refer to the online help for this function for more information.

### 3.4.1 uSYSInquire

This function (shown in FIGURE 3.3, *uSYSInquire function*) returns information about the connected UDAS unit. It allows you to verify or retrieve the UDAS serial number, identification, and version number of UDAS hardware and on-board firmware.

**FIGURE 3.3 uSYSInquire function**

#### Accessing the Help Page for this Function

Access the online Help manual as described in section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Inquire Functions* item and select **uSYSInquire**.

## 3.5 Function Sequence

When creating a diagram for an application, there is a sequence for the functions to follow. The recommended general function sequence is summarized below.

1. **uDeviceOpen.vi** is always the first U-Link function in any diagram being created. This function initializes the UDAS unit. Wire the device handle output from this function to all subsequent UDAS I/O function device inputs in the diagram. If more than one UDAS unit is connected to the host computer, multiple calls to **uDeviceOpen.vi** may be required to open a particular UDAS unit.
2. Although this is an optional function, including the **uSYSInquire.vi** function in each diagram is recommended because it enables you to verify the UDAS serial number, identification, and version number of UDAS hardware and on-board firmware.
3. The desired data acquisition processes can be performed at this point in the creation of the diagram.
4. After completing the I/O functions (such as uAIHSConfigureList, uAIHSEnable, and so forth), for this particular UDAS unit, insert the **uDeviceClose.vi** in the diagram to de-initialize the UDAS device.

### 3.5.1 Function Execution Sequencing

For a LabVIEW UDAS application to function properly, each function VI must execute in proper sequence. The sequence is determined by the connection of error code output and error code inputs terminals that exist on all VI function blocks. Error code inputs and outputs report errors and force the execution order of UDAS Support functions. Before a function block can execute, it must receive the error code information from the previous block. For information on error codes, see section 3.6.1, *Error Code Input and Error Code Output Terminals* and Appendix A, *Error Codes*.

When designing a function diagram, there is a sequence to follow. Open and examine the files in the UDAS Samples directory. The diagram flow always begins with **uDeviceOpen**, followed by **uSYSInquire**, the other functions that you select, and ends with the **uDeviceClose** function.

## 3.6 Sample Program

As an illustration of the relationship between a U-Link driver-based program and one created with the LabVIEW UDAS Support software, a partially listed driver-based code program and its UDAS Support VI equivalent is included in this chapter.

#### Driver-Based Sample Program

...UDAS Initialization steps:

error_code = **uDeviceOpen** (&myUDASHandle);
ErrorRoutine (0, error_code, "Error during uDeviceOpen");

...Configuration steps:

error_code = **uAIConfigureList** (myUDASHandle, 1, ai_list);
ErrorRoutine (0, error_code, "Error in uAIConfigureList");

...Data Acquisition steps:

error_code = **uAIRead** (myUDASHandle, 0, 0, 1, &counts);
ErrorRoutine (0, error_code, "Error in uAIRead");

...Data Processing and display code steps, etc.

...UDAS Initialization steps:

**uDeviceClose** (myUDASHandle);

### Explanation of the Sample Program Driver-Based Code

The **uDeviceOpen** in the first code section shown above initializes a UDAS unit.

Data acquisition process is being configured in the second section of code. **uAIConfigureList** configures a list of analog input channels on a UDAS unit.

The actual acquisition process is performed in the third section. The **uAIRead** function reads an analog input channel, returning the data value in counts. The utility function **uCountsToVolts** converts integer counts data to a floating point voltage value.

The **uDeviceClose** in the last code section de-initializes the UDAS unit.

FIGURE 3.4, *Diagram of a Sample-Based Program*, shows an actual LabVIEW-based application that was created using LabVIEW UDAS Support VIs. In comparing the Diagram with the code segments listed on the previous page, note the similarities between the U-Link driver functions and UDAS Support VIs.

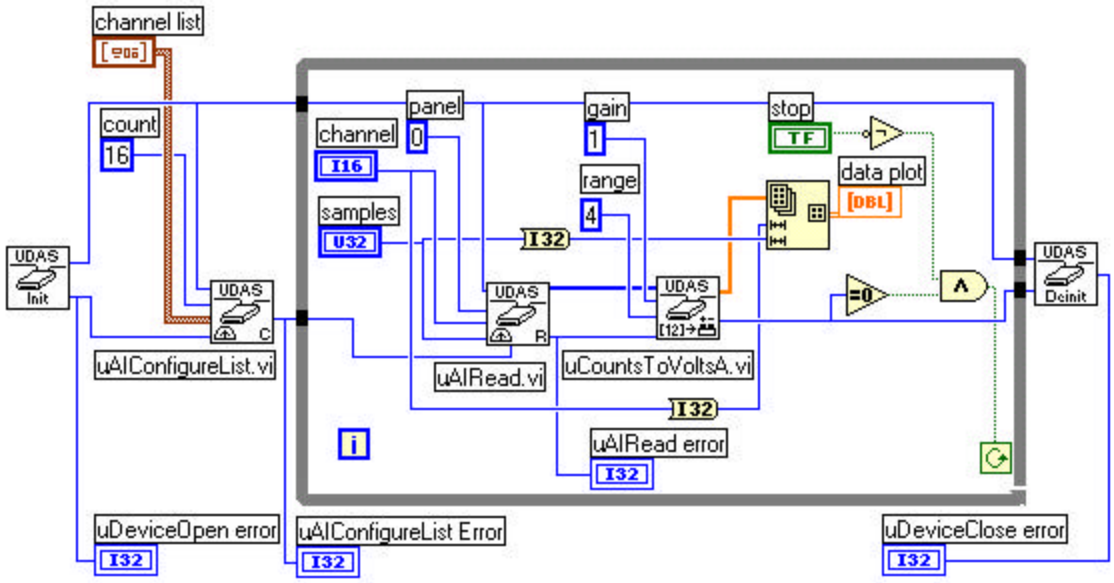



**FIGURE 3.4 Diagram of a Sample-Based Program**

FIGURE 3.5, *Front Panel Snapshot from a Driver-Based Sample Program*, displays analog data on a chart.

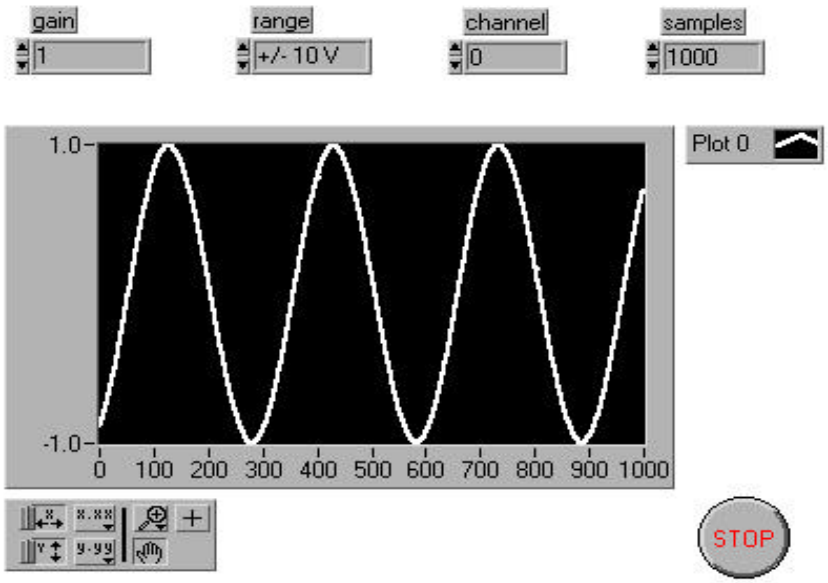



**FIGURE 3.5 Front Panel Snapshot from a Driver-Based Sample Program**

## 3.6.1 Error Code Input and Error Code Output Terminals

All UDAS Support VIs have an error code input terminal. The error code output from a LabVIEW UDAS Support VI connects to the error code input terminal of the UDAS Support VI that executes next. This code indicates the success or failure of the function. A zero output indicates that the function successfully completed without having any errors occur.

If the error code value is greater than zero, a critical error has occurred. There are no calls made to the U-Link DLL (**I3USB.DLL**), except for **uDeviceClose**, and the error code is passed to the error code output. This error code output terminal is connected to the next UDAS Support VI's error code input. The error passes from one UDAS Support VI to the next until the last UDAS Support VI in the application is reached and it stops, unless the diagram is set up to run in a continuous loop.

> **Note:** Avoid using the abort button (red octagon shaped button) to stop a program from running. Instead, build Stop buttons into your function diagram.

When a critical error occurs, calls are not made to the U-Link DLL. The example of **uAIRead.vi** is shown in FIGURE 3.7, *Error Code Boolean Case Structure (False)* and FIGURE 3.6, *Error Code Boolean Case Structure (True)*.

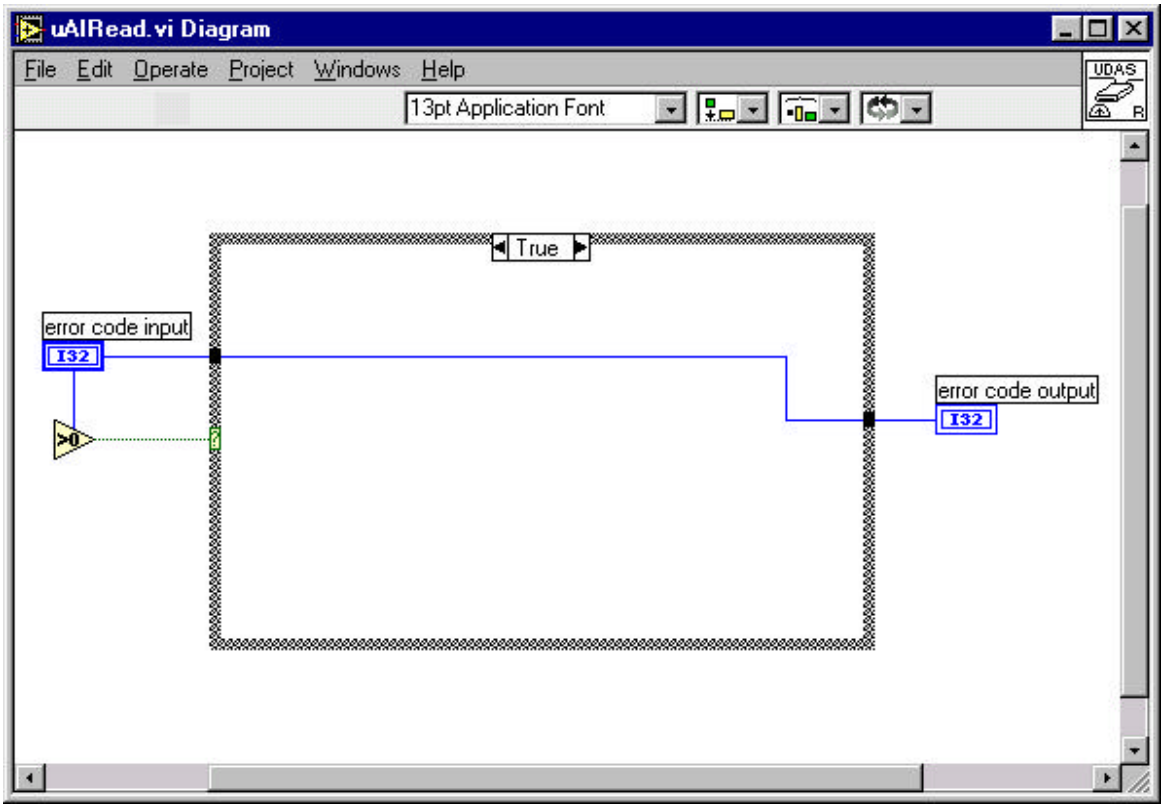



**FIGURE 3.6 Error Code Boolean Case Structure (True)**

The UDAS Support Library contains all of the error codes. For a quick reference of error codes, see Appendix A, *Error Codes*, section A.2, *List of Error Codes*.

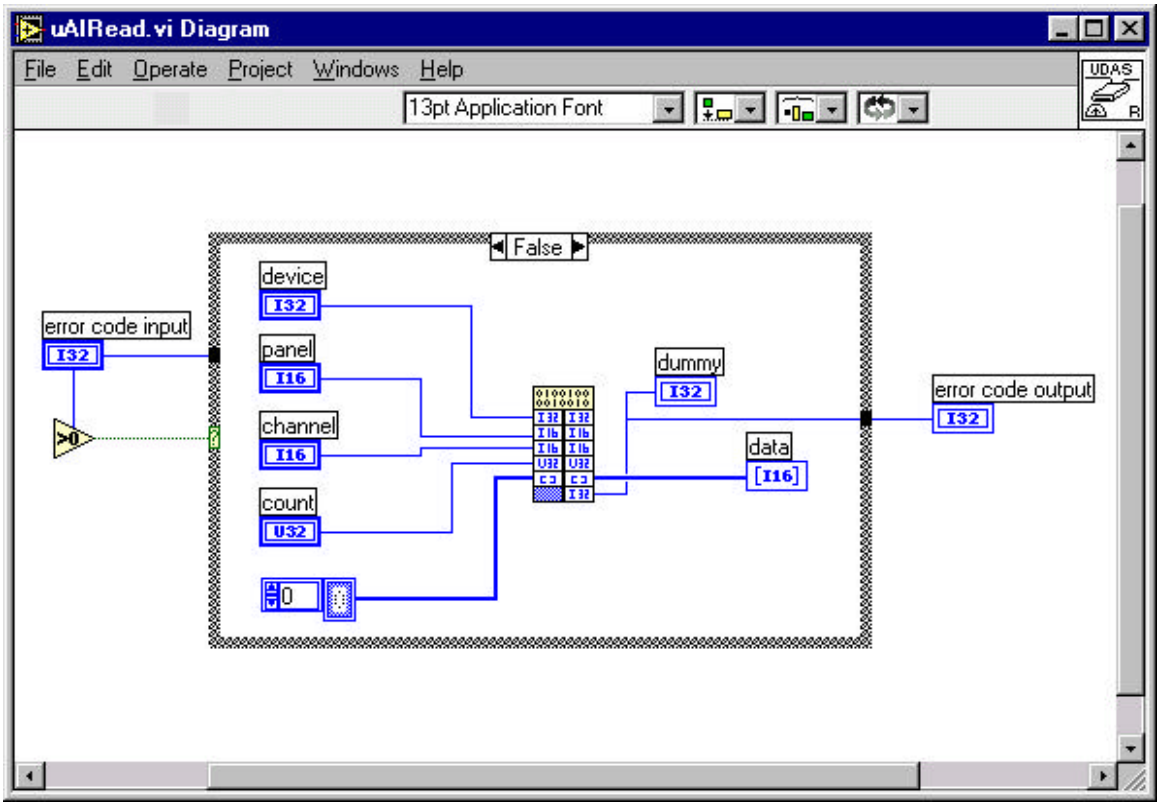



**FIGURE 3.7 Error Code Boolean Case Structure (False)**

## 3.7 Copying Examples as the Basis of Applications

It is easy to expand or modify an existing diagram to create a custom application. Make a copy of any of the sample files in the UDAS Samples directory and customize it for your application.

The UDAS Samples directory contains examples of applications from each functional category:

- Analog Input
- High-Speed Analog Input
- Analog Output, Digital I/O (port and bit)
- High-Speed Analog Output
- Hardware Counter
- Continuous High-Speed Analog Input
- Thermocouples

All of the sample programs make judicious use of indicators for error codes. When developing an application based on an example, displaying the error information makes it easier to debug and track errors. Error code displays are not required in the final application. When creating special data clusters or arrays for input to the more complex UDAS function VIs, it will save time by copying and pasting the structure needed from the sample into your application.

The UDAS Examples folder contains sample files for each type of function as shown in FIGURE 3.8, *UDAS Function Sample Files*.

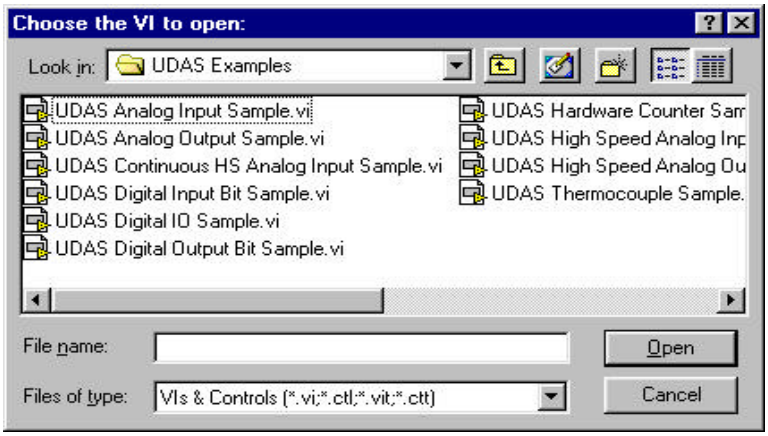



**FIGURE 3.8 UDAS Function Sample Files**

**UDAS Function Sample files include:**

UDAS Analog Input Sample.vi
UDAS Analog Output Sample.vi
UDAS Continuous HS Analog Input Sample.vi
UDAS Digital Input Bit Sample.vi
UDAS Digital IO Sample.vi
UDAS Digital Output Bit Sample.vi
UDAS Hardware Counter Sample.vi
UDAS High Speed Analog Input Sample.vi
UDAS High Speed Analog Output Sample.vi
UDAS Thermocouple Sample.vi

A UDAS low-level VI does not start the execution until the error code information is processed and passed by the previous UDAS VI function. If the error code indicates that a critical error has occurred (a code greater than zero, the UDAS VI receiving the error does not perform its function (except for the **uDeviceClose** function). The UDAS VI passes the error code, at its error code input, directly to its error code output.

This process repeats for the next UDAS VI, and continues until it reaches the last UDAS VI in the application, preventing the application from executing beyond the point at which the critical error occurred. Refer to the *UDAS Support Library online manual* for detailed information.

The **uDeviceOpen** VI must execute before any other LabVIEW UDAS Support function can be successfully executed. **uDeviceOpen** initializes the U-Link DLL and allocates the system resources required by the software.

The **uDeviceClose** VI deinitializes the U-Link drivers, freeing the resources the software uses before it exits the program. For more information on initialization and deinitialization functions, and system inquiry functions, refer to the online help pages for these functions.

> **Important:** The execution order of the UDAS VIs is determined by the connection of error code inputs and outputs in the LabVIEW UDAS application.

# 3.8 UDAS Support Library Functions

### Accessing the UDAS Support Library from LabVIEW

The UDAS Support Library functions in LabVIEW are accessed from the Functions Palette in the Diagram window. If the palette is not displayed, select the *Windows* sub-menu item, then select the *Show Diagram* sub-menu item. Select the *Show Functions palette* menu item to open the palette shown in FIGURE 3.9, *LabVIEW Functions Palette*.

**FIGURE 3.9 LabVIEW Functions Palette**

1. Click the user libraries icon to access User Libraries Palette shown in FIGURE 3.10, *User Libraries Palette.*

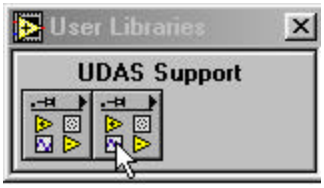



**FIGURE 3.10 User Libraries Palette**

2. Click the User Libraries icon that opens the UDAS Support Palette.

3. To select an icon in the UDAS Support Palette (shown in FIGURE 3.11, *UDAS Support Palette*), click on the desired function icon and drag the icon into the desired location in the diagram.

4. Complete and save the diagram for your application.

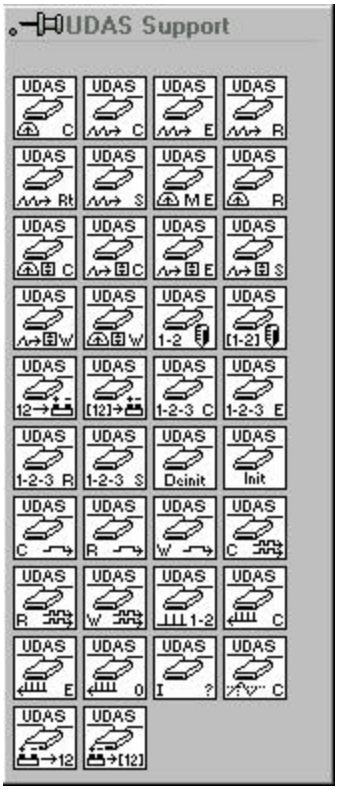



**FIGURE 3.11 UDAS Support Palette**

# Chapter 4 Support Library: Analog Input & High-Speed Analog Input Functions

## 4.1 Analog Input Functions

This section describes how to acquire analog data with a UDAS unit using the LabVIEW UDAS Support Library.

The two categories of functions provided are:

- Functions for software controlled analog input data acquisition
- Functions for hardware controlled high-speed analog input data acquisition

Software controlled acquisition allows the capturing of analog input data from one input channel, or all input channels, under software control. High-speed acquisition allows evenly spaced sampling of one or more inputs using a time-based source and triggering options. Data can be acquired from either one channel or all channels on the UDAS device. The addition of an AI-MAX 256 channel expansion system will allow collection of the data from up to 256 channels (up to 16 termination panels).

There are three analog input functions: **uAIConfigureList**, **uAIRead**, **uAIMAXEnable**. Each one of these functions performs a different task for analog input process as described below.

**uAIConfigureList.vi** configures, reconfigures, or unconfigures a list of analog channels for software-controlled analog data acquisition.

**uAIRead.vi** reads one or more previously configured analog input channels.

**uAIMAXEnable.vi** enables the specified termination panels on AI-MAX 256 channel analog input expansion system.

### Sample Program

The functions used for analog input applications can be arranged as shown in the sample program UDAS **Analog Input Sample.vi**.

### Using AI-MAX

If AI-MAX (256-channel analog input expansion system) is connected, the **uAIConfigureList** function has the capability to configure a maximum of 256 single-ended analog input channels. A maximum of 16 termination panels can be connected to AI-MAX. Use the **uAIMAXEnable** function to enable the termination panels to be used in the application. The **uAIListType** cluster contains a panel control which indicates the panel to be used. If a particular termination panel is not specified as active by the **uAIMAXEnable** function, error code 31, *AI PANEL ERROR* occurs.

### Using the Functions in the Diagram

- Begin with the initialization function **uDeviceOpen**.
- For analog input applications, the **uAIConfigureList** function configures a list of analog input channels, such as gain and range. This function must be placed in the diagram before input data can be read using the **uAIRead** function. Failure to follow this sequence results in error code 36, *AI NOT CONFIGURED ERROR* from the UDAS unit.
- If AI-MAX is being used in the application, the **uAIMAXEnable** function is required in the diagram and should be called before **uAIConfigureList**.
- Insert the other desired functions in the diagram for the application.
- The last UDAS Support function to use in the application is the **uDeviceClose.vi**, which deinitializes (closes the handle) to the UDAS unit.

### 4.1.1 uAIConfigureList

This analog input function (shown in FIGURE 4.1, *uAIConfigureList Function*) configures a list of analog input channels (such as gain range). See the online help page for detailed information about this function.

The channel list is specified through two VI inputs: *channel list* and *count*. The channel *list* input uses an analog input channel list LabVIEW cluster called **uAIListType**. For a specified analog input channel from a specified termination panel (panel control value should always be 0 if there is no AI-MAX connected), the **uAIListType** cluster allows you to set the gain, input range, single-ended or differential mode, and a zero-reference flag.

Use the *count* input to tell the function the number of channels to configure (maximum of 256 with AI-MAX and 16 without AI-MAX). The list that you create is an array of *count* elements, of cluster **uAIListType**.

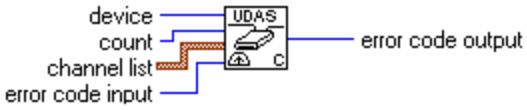



**FIGURE 4.1 uAIConfigureList Function**

#### Accessing the Help Page for this Function

Access the online Help manual as described in section FIGURE 3.2, *uDeviceClose function*. In the Contents dialog, double-click the *Analog Input Functions* item and select **uAIConfigureList**.

## 4.1.2 uAIRead

The **uAIRead** function (shown in FIGURE 4.2, *uAIRead Function*) follows the **uAIConfigureList** function and reads a previously configured analog input channel or all analog input channels. If AI-MAX is not used, the *panel* input should always be set to 0. If AI-MAX is used, then there are two methods for specifying the panel to be read:

- Either as a panel number, 0 - 15, or
- Use the code (-1) that indicates reading from all panels on the AI-MAX.

> **Note:** The panels must be enabled first via the uAIMAXEnable call.

On *channel* input, there are two methods for specifying the channel to be read:

- As channel number 0 - 15, or
- Using code (-1) indicates a reading from all channels on the specified termination panel.

The acquired data is returned in common format counts. Use the **uCountsToVolts.vi** or **uCountsToVoltsA.vi** functions to convert counts data to floating point voltage values.

The data array will be properly allocated in the event that more than one sample is returned, or if all channels (-1) or all panels (-1) are read. **uAIRead** automatically allocates the single-dimension array of sufficient size. It is important for you to know how the array size is determined and how the returned data is organized.

- If all-channels (-1) is specified, the returned data array allocates space for 16 channels, even though fewer channels may be configured.
- If all panels (-1) is specified, the returned data array allocates space for 16 panels with one or 16 channels each (depending on channel specification) even though not all panels are present.
- Multiplying by the number of samples per channel further increases the size of the data area.

See the online help page for detailed information about this function.

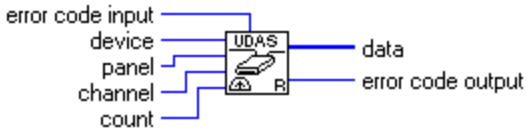



**FIGURE 4.2 uAIRead Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Analog Input Functions* item and select **uAIRead**.

## 4.1.3 uAIMAXEnable

This function enables specified termination panels on AI-MAX 256 channel AI expansion system. The **uAIMAXEnable** function (shown in FIGURE 4.3, *uAIMAXEnable Function*) is always placed in the diagram so that it is called before any analog input functions, or any digital I/O functions that reference Port 1.

If an AI-MAX controller is being used in the application, it uses bits 4 through 7 of digital Port 1. The *count* input is specified by the number of termination panels to be installed in the AI-MAX system. This value must be between 1 and 16. The *panels input* requires an array of index numbers of active panels (up to 15). To enable panel #3, set *count* to 1 and pass a one-element array containing the value 3 to this function. See the online help page for detailed information about this function.

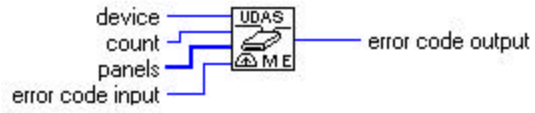



**FIGURE 4.3 uAIMAXEnable Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Analog Input Functions* item and select **uAIMAXEnable**.

## 4.2 High-Speed Analog Input Functions

A high-speed analog input process uses a pacer, either an internal rate generator or an externally supplied signal, to time the acquisition of analog samples. One or more channels may be sampled. The process may be started by a trigger event or under program control. It may be stopped after acquiring a set amount of data, after a trigger event, or under program control. It may be stopped after acquiring a set amount of data, after a trigger event, or under program control.

There are five high-speed analog input functions: **uAIHSConfigureList**, **uAIHSEnable**, **uAIHSRead**, **uAIHSReturn**, and **uAIHSStatus**. Each one of these functions performs a different task for a high-speed analog input process as described below.

### Functions

**uAIHSConfigureList.vi** configures a high-speed analog input process.

**uAIHSEnable.vi** enables or disables a high-speed analog input process.

**uAIHSRead.vi** reads the data acquired in a high-speed analog input process and returns it to your application.

**uAIHSReturn.vi** returns data from device buffers to a host buffer managed by the driver.

**uAIHSStatus.vi** obtains the status and buffer information of high-speed analog input process.

### Additional Functions Required for High-Speed Data Acquisition

#### Rate Generator Functions

**uRGConfigure.vi** configures a rate generator channel with initial count value.

**uRGEnable.vi** enables or disables a rate generator channel.

**uFrequencyToRGCounts.vi** computes the count value used to program a rate generator channel for a specific frequency.

#### Trigger Function

**uTRIGConfigure.vi** configures an analog trigger channel for a high-speed analog input or output process.

## Using the Functions for High-Speed Analog Input

To configure a list of analog input channels for a high-speed analog process, use the function **uAIHSConfigureList**. This call also specifies the start and stop methods of the acquisition, pacer source and optional trigger source.

The channel list is specified to the **uAIHSConfigureList** call and uses the same format as a channel list specified to the **uAIHSConfigureList** call. An array of **uAIListType** clusters is passed to the **uAIHSConfigureList** call's *channel list* input.

After configuring the analog input channels and configuring the pacer and trigger sources (through appropriate U-Link function call), data from these analog input channels can be read by enabling the high-speed process using **uAIHSEnable**, then calling the **uAIHSReturn** and **uAIHSRead** function calls.

To obtain status information about the acquisition process, use the **uAIHSStatus** function.

## Sample Program

For more information on how to perform the high-speed acquisition of analog data, refer to the **UDAS High-Speed Analog Input Sample.vi**. In this sample, the data acquisition process will continue until a specified number of samples is acquired (Stop on Terminal Count mode). Another high-speed analog input sample, the **UDAS Continuous HS Analog Input Sample.vi**, illustrates how to acquire the analog data continuously. The data acquisition continues indefinitely, unless you click a Stop control to terminate the process (Stop on Command mode).

## Trigger Functions for High-Speed Analog Input

There is one trigger function, **uTRIGConfigure**, that configures an analog trigger channel for high-speed analog input on a UDAS unit. The analog trigger circuit on a UDAS unit uses any analog input channel as input to an internal comparator.

When the analog input trigger is used, analog output channel 1 should not be used for other purposes. Analog output channel 1 is used to generate a threshold voltage on the other input to the internal comparator circuit. When the input voltage on the selected input channel crosses the threshold voltage, a trigger signal is generated. See the UDAS Support online manual for more information on this function.

## Rate Generator Functions for HS Analog Input

Rate generator functions are used with analog input and analog output functions.

One rate generator's output is also available at the UDAS unit's I/O connector which is selected by the **uSetOption** function. Use the **uFrequencyToRGCounts** function to obtain the count value for the desired output frequency.

A rate generator's output frequency is the result of dividing the on-board 8 MHz timebase frequency by a 24-bit integer *count*. For additional information on rate generator functions, see the UDAS Support online manual.

### uFrequencyToRGCounts

The **uFrequencyToRGCounts** function (shown in FIGURE 4.4, *uFrequencyToRGCounts Function*) computes the count value used to program a rate generator channel for a specific frequency.

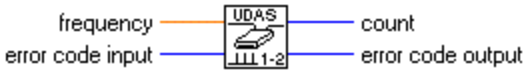



**FIGURE 4.4 uFrequencyToRGCounts Function**

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Utilities Functions* item and select **uFrequencyToRGCounts**.

### uRGConfigure

This rate generator function (shown in FIGURE 4.5, *uRGConfigure Function*) configures the analog input rate generator channel on a UDAS unit that has a count value stored in the buffer pointed to by data. Pulses generated by the analog input rate generator are used to start A/D conversions in high-speed analog input processes.

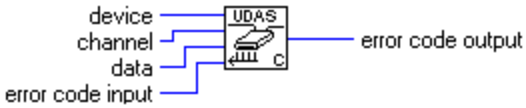



**FIGURE 4.5 uRGConfigure Function**

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Rate Generator Functions* item and select **uRGConfigure**.

### uRGEnable

This rate generator function (shown in FIGURE 4.6, *uRGEnable Function*) enables or disables a rate generator channel.

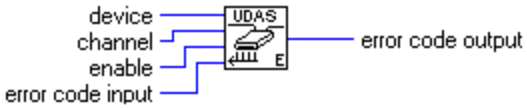



**FIGURE 4.6 uRGEnable Function**

To enable or disable a rate generator, use the **uRGEnable** function.

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Rate Generator Functions* item and select **uRGEnable**.

## 4.2.1 uAIHSConfigureList

This high-speed analog input function (shown in FIGURE 4.7, *uAIHSConfigureList Function*) configures a list of analog input channels. This function is also used to specify the start and stop modes of the acquisition, the pacer (timebase rate generator or external source), and trigger (analog input trigger, external trigger, or none).

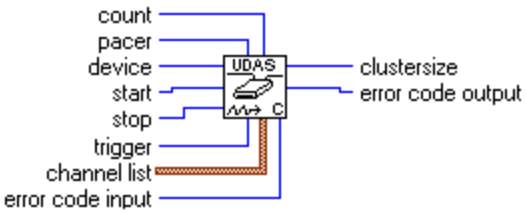



**FIGURE 4.7 uAIHSConfigureList Function**

The channels in the list must be specified in consecutive order, from channel 0 to N with no gaps, where N = the last channel in the list. Each channel in the list must have the same configuration settings for gain, input voltage range, and input type (single-ended or differential). A zero-reference cannot be used in high-speed acquisitions. Set the zero reference field of **uAIListType** cluster to 0.

The **uAIHSConfigureList** call returns a value in its *clustersize* output indicating the number of bytes required to store the input data acquired for one pass through the channel list. The information is used by the **uAIHSEnable** and **uAIHSRead** functions to allocate a data buffer in the host memory for returned data.

### Start and Stop Modes

High-speed start and stop methods are specified in the **uAIConfigureList** call's *start* and *stop* inputs.

Start Modes supported by the UDAS units include:

- Start on Command (numeric value of 0)
- Start on Trigger (numeric value of 1)

Supported Stop Modes include:

- Stop on Terminal Count (numeric value of 2)
- Stop on Trigger with Delay (numeric value of 1)
- Stop on Command (numeric value of 0)

The following paragraphs discuss the operation of each start and stop mode.

### Start on Command

When specifying the *Start on Command* mode in **uAIHSConfigureList**, data acquisitions begin when the **uAIHSEnable** function, with its enable parameter set to a numeric value of 1, is called.

### Start on Trigger

When specifying the *Start on Trigger* mode, data acquisition begins when the programmed trigger condition occurs (after the process is enabled by a call to **uAIHSEnable** with its enable parameter set to a numeric value of 1).

### Stop on Terminal Count

When specifying the *Stop on Terminal Count* mode, acquisitions stop when the requested number of samples per channel are captured.

- Use the **uAIHSStatus** or **uAIHSReturn** call to detect the end of a terminal count process.

### Stop on Command

When using the *Stop on Command* mode, data acquisitions stop when the **uAIHSEnable** function is called with its enable parameter set to a value of 0. For programs that contain a *Stop on Command* mode, data is acquired in a *circular buffer* on the host. The number of channels in the channel list (count parameter in the **uAIHSConfigureList**) is multiplied by the maximum number of samples to read per channel. This calculation is used to determine the size of this data buffer (count parameter in the **uAIHSEnable**).

- If the total amount of data acquired will exceed 64K bytes, use the **uAIHSReturn** call to return data to the host during the process. Data is written to a circular buffer until the process stops.

### Stop on Trigger with Delay

Acquisitions stop when the delay count is reached following the trigger event. Data is acquired to a circular buffer until the process stops. If the total amount of data acquired is to exceed 64K bytes, use the **uAIHSReturn** call to return data to the host during the process. Data is written to a circular buffer until the process stops.

> **Important:** Unless you plan to configure a process once and restart it multiple times, free the local data buffer by calling **uAIHSEnable** with its enable parameter set to 0 and its free parameter set to logic 1 when the process terminates. If using the Stop on Command option, do not free the buffer until the data has been read from the buffer.

For example, for a digital oscilloscope application using a process configured for *Start on Trigger* and *Stop on Terminal Count* modes, call **uAIHSConfigureList** once, followed by repeated calls to **uAIHSEnable** and **uAIHSRead**. When the process is no longer needed, call **uAIHSEnable** to terminate the process and free the UDAS unit's data buffer.

### Pacer Input

High-speed analog input data acquisition requires the use of a pacing signal or timebase. Pacing signal pulses initiate A/D conversions at regularly spaced intervals. The *pacer* input is used to select a pacer source.

To use on-board analog input rate generator, set this input to 0. To use an external pacer source, set the input to 1. If an external pacer source is selected, then a TTL-level source clock must be connected to an External Input (at pin 47 of the UDAS unit's I/O connector). Use the **uRGConfigure** call to configure analog input rate generator with a desired count value. The **uFrequencyToRGCounts** function can be used to obtain the count value for the desired output frequency. Call **uRGEnable** to enable or disable the rate generator. For additional information on this function, refer to the online help for this function in the UDAS Support Library online manual.

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Analog Input (High-Speed) Functions* item and select **uAIHSConfigureList**.

## 4.2.2 uAIHSEnable

This high-speed analog input function (shown in FIGURE 4.8, *uAIHSEnable Function*) is used to enable (start) or disable (stop) a high-speed process.

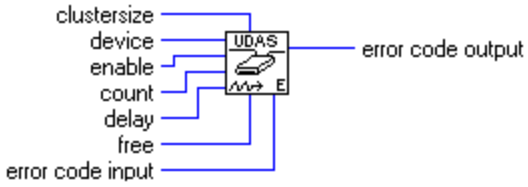



**FIGURE 4.8 uAIHSEnable Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Analog Input (High-Speed) Functions* item and select **uAIHSEnable**.

## 4.2.3 uAIHSReturn

The next function to include in the diagram is **uAIHSReturn**. This function (shown in FIGURE 4.9, *uAIHSReturn Function*) returns the data acquired in a high-speed analog input process from device buffers to host buffers. This call also obtains the status and current buffer information of the high-speed analog input process.

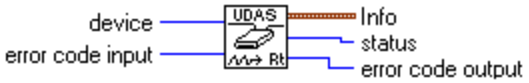



**FIGURE 4.9 uAIHSReturn Function**

The data is transferred from the UDAS unit to the host computer via the USB. Actual data from the buffer can be obtained by calling **uAIHSRead**.

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Analog Input (High-Speed) Functions* item and select **uAIHSReturn**.

## 4.2.4 uAIHSRead

This high-speed analog input function (shown in FIGURE 4.10, *uAIHSRead Function*) reads the data acquired in a high-speed analog input process. This call also returns information about the data buffer (*info*).

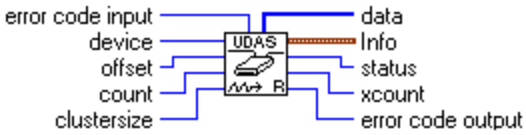



**FIGURE 4.10 uAIHSRead Function**

The **uAIHSRead** call returns data from the buffer accumulated on the *host* by the running data acquisition process. If the data requested has not been returned by the UDAS device, this function may check the status to read back any available data.

If the buffer does not contain the number of clusters requested, the function calls **uAIHSReturn** first. The status returned is the status corresponding to the most recent call to **uAIHSReturn** or **uAIHSStatus**.

The data output contains an array of received data. Analog input data is returned in the following format: sample 0 panel 0 channel 0, sample 0, panel 0, channel 1, sample 0 panel 0 channel 2, ...,sample 0 panel 0 channel 15, sample 0 panel 1 channel 0, sample 0 panel 1 channel 1,...,sample 1 panel 0 channel 0, sample 1 panel 0 channel 1,..., sample 1 panel 1 channel 0, and so forth.

The **uCountsToVolts** or **uCountsToVoltsA** function can be used toconvert common format data to floating point voltage values, if desired.

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Analog Input (High-Speed) Functions* item and select **uAIHSRead**.

## 4.2.5 uAIHSStatus

The **uAIHSStatus** function (shown in FIGURE 4.11, *uAIHSStatus Function*) obtains the status and current buffer information of a high-speed analog input process. This function is useful for monitoring the system when a *Stop on Terminal Count* process is performed, or when a trigger has occurred.

**FIGURE 4.11 uAIHSStatus Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.2, *Function Wiring Connection Information*. In the Contents dialog, double-click the *Analog Input (High-Speed) Functions* item and select **uAIHSStatus**.

# Chapter 5 Support Library: Analog Output & High-Speed AO Functions

## 5.1 Analog Output Functions

This chapter describes how to output analog data with a qualified UDAS unit using the LabVIEW UDAS Support Library. Two categories of functions are provided. One function group allows a user to write a single piece of data at a time to either one or two analog output channels under software control. Another function group allows a user to write a continuous stream of data to one or both analog output channels under hardware control. This feature can be used to generate a limited waveform on selected output channel(s).

### Functions

There are two analog output functions, **uAOConfigureList** and **uAOWrite**.

**uAOConfigureList.vi** configures a list of analog output channels.

**uAOWrite.vi** writes data to previously configured analog output channels.

To configure a list of analog output channels, use the **uAOConfigureList** function. Once configured, use the **uAOWrite** function to write data to these analog output channels. For more information on how to write the data to these analog output channels, refer to the **UDAS Analog Output Sample.vi**.

### 5.1.1 uAOConfigureList

This analog output function (shown in FIGURE 5.1, *uAOConfigureList Function*) configures a list of analog output channels on a UDAS unit. Before data can be written to analog output channels, they must be configured.

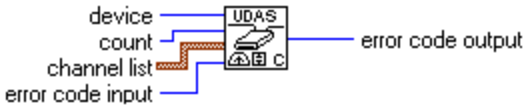



**FIGURE 5.1 uAOConfigureList Function**

The channel list is specified through two VI inputs: *channel list* and *count*. The *channel list* input uses an analog output channel list LabVIEW cluster called **uAOListType**. Use the *count* input to

tell the function the maximum number of channels to configure (maximum of 2). The list that you create is an array of *count* elements, of cluster **uAOListType**.

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.8, *UDAS Support Library Functions*. In the Contents dialog, double-click the *Analog Output Functions* item and select **uAOConfigureList**.

## 5.1.2 uAOWrite

This analog output function (shown in FIGURE 5.2, *uAOWrite Function*) writes data to a previously configured analog output channel, or to all channels. The counts data buffer must be properly dimensioned and initialized for the number of channels to be used in the application.

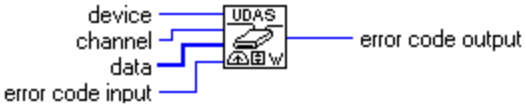



**FIGURE 5.2 uAOWrite Function**

On the *channel* input, there are two methods for specifying the channel to be written:

- Either as channel number 0 or 1, or
- Use the code (-1) that indicates writing to both channels.

The *data* array holds the data to be written to the specified analog output(s). Specify the data values in counts. Use the **uVoltsToCounts.vi** or **uVoltsToCountsA.vi** functions to convert floating point voltage values to counts data.

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.8, *UDAS Support Library Functions*. In the Contents dialog, double-click the *Analog Output Functions* item and select **uAOWrite**.

## 5.2 High-Speed Analog Output Functions

There are four high-speed analog output functions, **uAOHSConfigure**, **uAOHSEnable**, **uAOHSStatus**, and **uAOHSWrite**. Each one of these functions performs a different task for high-speed analog input process as described below.

### Functions

**uAOHSConfigureList.vi** configures a high-speed analog output process.

**uAOHSEnable.vi** enables the analog output process and transmits the data to be written, or disables the analog output process.

**uAIHSStatus.vi** obtains the status of a high-speed analog output process.

**uAOHSWrite.vi** downloads a segment of a continued analog output waveform to a UDAS unit.

### Additional Functions Required for a High-Speed Data Acquisition Process

#### Rate Generator Functions

**uRGConfigure.vi** configures a rate generator channel with initial count value.

**uFrequencyToRGCounts.vi** computes the count value used to program a rate generator channel for a specific frequency.

#### Trigger Function

**uTRIGConfigure.vi** configures an analog trigger channel for a high-speed analog input or output process.

### Sample Program

Additional information on how to perform the data acquisition on the port level, refer to the **UDAS High-Speed Analog Output Sample.vi**.

The high-speed analog output process occurs with the following steps.

1. Disable any previously configured high-speed analog output via **uAOHSEnable**.
2. Configure output channels, pacer and trigger using **uAOHSConfigureList**.
3. If the pacer is the on-board rate generator, set its parameters via **uRGConfigure**.
4. To synchronize with a high-speed data acquisition process, set the parameters of the data acquisition process, its pacer and trigger.

5. Start the data output process via **uAOHSEnable**. This call enables the rate generator and transmits initial data to the output process.
6. Start the data acquisition process if it was configured in step 4.
7. If a longer waveform is needed, call **uAOHSWrite** to transmit additional data to the output process.
8. Check the status of the high-speed analog output process using **uAOHSStatus**.
9. To stop writing waveform to analog channel(s), call **uAOHSEnable**. If the repeat mode is set to 0, the output process will stop automatically.

The initial data is transmitted via the **uAOHSEnable** call. Additional waveform data can be sent by using the **uAOHSWrite** function. The **uAOHSConfigureList** function returns a value in its cluster size parameter, which is the number of bytes to be allocated for the data to be transmitted for one pass through the channel list. The clustersize parameter value must be passed to the **uAOHSWrite** and **uAOHSEnable** functions. The clustersize value can also be used to determine the local buffer size required to be allocated for analog output data by the **uAOHSWrite** and **uAOHSEnable** functions. See the online help pages for more information on these functions.

### Configuring an Analog Output Rate Generator Channel

The **uRGConfigure** function configures the analog output rate generator channel on a UDAS unit that has a count value stored in the buffer pointed to by data. Pulses generated by the analog output rate generators are used to update analog output channels.

One rate generator's output is also available at the UDAS unit's I/O connector which is selected by the **uSetOption** function. Use the **uFrequencyToRGCounts** function to obtain the count value for the desired output frequency.

### Trigger Functions for High-Speed Analog Output

There is one trigger function, **uTRIGConfigure**, that configures an analog trigger channel for analog output on a UDAS unit. The analog trigger circuit on a UDAS unit uses any analog input channel as input to an internal comparator.

Analog output channel 1 is used to generate a threshold voltage on the other input to the internal comparator circuit. When the input voltage on the selected input channel crosses the threshold voltage, a trigger signal is generated. When the analog input trigger is used, analog output channel 1 should not be used for other purposes. See the UDAS Support online manual for more information on this function.

The last UDAS Support function to use in the application is the **uDeviceClose.vi**, which deinitializes (closes the handle) to the UDAS unit.

## Rate Generator Functions for High-Speed Analog Output

### uRGConfigure

This rate generator function (shown in FIGURE 5.3, *uRGConfigure Function*) configures the analog output rate generator channel on a UDAS unit that has a count value stored in the buffer pointed to by data. Pulses generated by the analog output rate generators are used to update analog output channels.

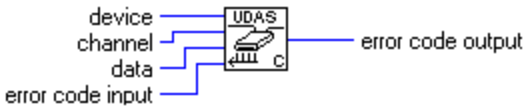



**FIGURE 5.3 uRGConfigure Function**

A rate generator's output frequency is the result of dividing the on-board 8 MHz timebase frequency by a 24-bit integer *count*. For additional information on rate generator functions, see the UDAS Support online manual.

## Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*, section 3.8, *UDAS Support Library Functions*. In the Contents dialog, double-click the *Rate Generator Functions* item and select **uRGConfigure**.

## 5.2.1 uAOHSConfigureList

This high-speed analog output function (shown in FIGURE 5.4, *uAOHSConfigureList Function*) configures a high-speed analog output process on a UDAS unit. A list of the channels and channel parameters, and various high-speed parameters are specified to the call to configure the high-speed process. The channel list uses an array of clusters of **uAOListType**, as the **uAOConfigureList** function. The **uAOHSConfigureList** high-speed parameters allow a pacer source and trigger source to be specified.

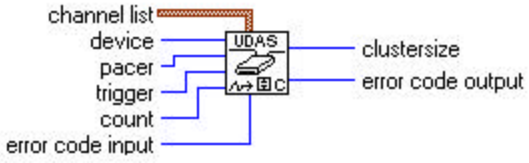



**FIGURE 5.4 uAOHSConfigureList Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Analog Output (High-Speed) Functions* item and select **uAOHSConfigureList**.

## 5.2.2 uAOHSEnable

This high-speed analog output function (shown in FIGURE 5.5, *uAOHSEnable Function*) enables or disables a high-speed analog output process, transmits the data to output.

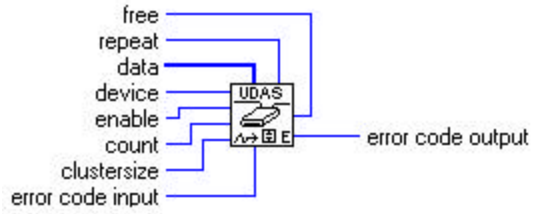



**FIGURE 5.5 uAOHSEnable Function**

### Enabling or Disabling a High-Speed Analog Output Process

To enable or disable a high-speed analog output process that transmits data to the output, use the **uAOHSEnable** function.

If the data array passed is empty, the command resumes the output of a waveform that has already been transferred to the UDAS output buffer. If a waveform is not present, error **AOHS_DATA_ERR** (60) occurs.

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Analog Output (High-Speed) Functions* item and select **uAOHSEnable**.

## 5.2.3 uAOHSStatus

This high-speed analog output function (FIGURE 5.6, *uAOHSStatus Function*) obtains the status information for a high-speed analog output process. The library bit masks for testing particular status conditions are located on the online help page for this function.

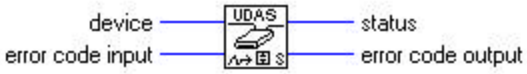



**FIGURE 5.6 uAOHSStatus Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Analog Output (High-Speed) Functions* item and select **uAOHSStatus**.

## 5.2.4 uAOHSWrite

This high-speed analog output function (shown in FIGURE 5.7, *uAOHSWrite Function*) downloads a segment of a continued output waveform to the device.

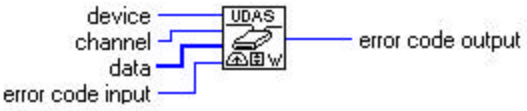



**FIGURE 5.7 uAOHSWrite Function**

### Downloading a Segment of a Continued Analog Output Waveform

To download a segment of a continued analog output waveform to the UDAS device, use the **uAOHSWrite** function. The output will terminate if the UDAS unit runs out of data before more data is sent. In order to have continuous operation, it is necessary to have a steady flow of data.

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Analog Output (High-Speed) Functions* item and select **uAOHSWrite**.

# Chapter 6 Support Library: Digital I/O Port & Bit Functions

## 6.1 Introduction

This chapter discusses the use of digital I/O port and bit functions. There are two digital I/O ports of 8 bits each on a UDAS unit. Digital I/O port functions can be used to read or write a group of related signals as a data byte. To read or write individual unrelated signals separately (not as a group), or to use a digital input signal as a counter or a trigger, use digital I/O bit functions.

## 6.2 Digital Input/Output Port Functions

### Functions

**uDIOConfigureList** configures a list of digital I/O ports on a UDAS unit.
**uDIORead** reads the digital I/O data from the ports.
**uDIOWrite** writes data to the digital output ports.

For more information on how to perform data acquisition on the port level, refer to the **UDAS Digital I/O Sample.vi**.

### 6.2.1 uDIOConfigureList

This digital input/output port function configures a list of digital I/O ports. This function must be used to configure Digital I/O ports *before* input or output data can be read from, or written to these ports. If an AI-MAX controller is being used, it uses bits 4 through 7 of port 1. Port 1 may only be configured as an output.

The port list is specified through two VI inputs, *port list* and *count.* Either port can be configured independently as an input or an output. The *port list* input uses a digital I/O port list LabVIEW cluster called **uDIOListType**. Use the *count* input to tell the function the number of ports to configure (maximum of two for a UDAS unit). The list that you create is an array of *count* elements, of **uDIOListType**.

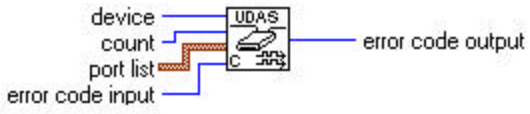



**FIGURE 6.1 uDIOConfigureList Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Digital Input/Output Port Functions* item and select **uDIOConfigureList**.

## 6.2.2 uDIORead

This digital input/output port function reads one or two digital ports configured for input.

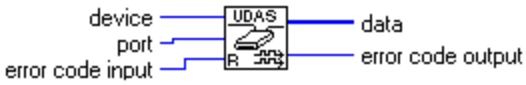



**FIGURE 6.2 uDIORead Function**

To read one or two digital I/O ports configured for input, use the **uDIORead** function. If the port was configured for output, an error is returned. The data buffer must be dimensioned properly for the number of ports to be read.

There are two methods for specifying the port to be read or written:

- Either as port number, 0 or 1, or
- Use the code (-1) that indicates access to both port 0 and port 1

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Digital Input/Output Port Functions* item and select **uDIORead**.

## 6.2.3 uDIOWrite

This digital input/output port function writes data to one or two digital ports configured for output. The port(s) must be configured for output operation and the data buffer must be dimensioned and properly initialized for the number of ports. Refer to section 6.2.2, *uDIORead* for more information on port input.

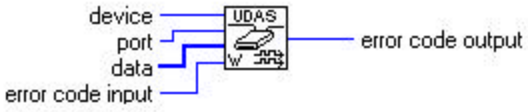



**FIGURE 6.3 uDIOWrite Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Digital Input/Output Port Functions* item and select **uDIOWrite**.

## 6.3 Digital Input/Output Bit Functions

There are three digital input/output bit level functions: **uDIOBITConfigureList**, **uDIOBITRead**, **and uDIOBITWrite**.

### Functions

**uDIOConfigureList** configures of a list of digital I/O ports on a UDAS unit.

**uDIORead** reads the digital I/O data from the ports.

**uDIOWrite** writes data to the digital output ports.

### Setting the Values for a Digital Output Port Bit(s)

The **uDIOBITWrite** function sets the value(s) for a digital output port bit or bits. To use this function, the port bit must have been configured Digital Output Bit mode (0) using the **uDIOBITConfigureList** function. If the port bit is configured for one of the Trigger Input Modes (3, 7, 11), the **uDIOBITWrite** function should be used to arm or disarm the trigger. This function is not valid for bits configured for normal input mode (1).

## 6.3.1 uDIOBITConfigureList

This digital input/output bit function configures a list of digital I/O port bits. The port list is specified through two VI inputs: *port bit list* and *count*. The port list input uses a digital I/O port bit list LabVIEW cluster called **uDIOBITListType**. Use the *count* input to tell the function the number of port bits to configure (maximum of 16 for a UDAS unit). The list that you create is an array of count elements, of type **uDIOBITListType**.

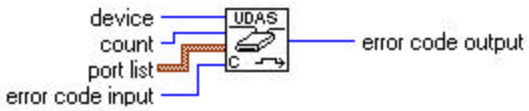



**FIGURE 6.4 uDIOBITConfigureList Function**

Digital I/O port bits must be configured using **uDIOBITConfigureList** before data can be read from, or written to, these bits. The initial data value is written to output bits immediately after they are configured as outputs. If AI-MAX is being used, it uses bits 4 through 7 of port 1. In this case, port 1 can only be configured for output using bits 0 through 3.

## Descriptions of Input and Output DIO Modes

### Input Mode

The input mode allows direct (unlatched) reading of a digital input bit. A read of a bit configured for normal mode returns the current level (1 or 0) of the input signal. The LabVIEW numeric value is 1.

### OutputMode

Writing to a bit configured for output mode sets the bit's output level 1 or 0. The LabVIEW numeric value is 0.

### High-to-Low or Low-to-High Trigger Modes

These trigger modes on the UDAS device increment a counter on both high-to-low or low-to-high signal edge. The LabVIEW numeric value is 3 for high-to-low edge trigger mode, and 7 for low-to-high edge trigger mode.

### Any Mode (High-to-Low and Low-to-High Trigger Modes)

This trigger mode allows the UDAS device to increment a counter on both high-to-low and low-to-high signal edges. The LabVIEW numeric value is 11.

For additional information on using the various digital input modes, refer to the **UDAS Digital Input Sample.vi**. For more information on using a digital output bit mode, refer to the **UDAS Digital Output Bit Sample.vi**.

## Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Digital Input/Output Bit Functions* item and select **uDIOBITConfigureList**.

## 6.3.2 uDIOBITRead

This general purpose digital input/output bit function reads a digital input port bit configured for any input mode. This function reads a digital I/O port bit, selected port bits, or all port bits on a specified port, configured for either Normal Input Mode (1) or Trigger Input modes (3, 7, 11).

> **Note:** If the port bit is configured for Trigger Input Modes (3, 7, 11), the uDIOBITWrite function must be called before uDIOBITRead is called for the first time to arm the trigger. uDIOBITWrite must also be called after the port bit is reconfigured for trigger mode. See the uDIOBITWrite description in the Online Help page for more information.

Port bit input requires 2-bytes integer value. There are two methods for specifying the port bit to be read or written:

- Either as a port bit number, 0-7 (high byte is 0), or
- Use the high byte as a bit mask that indicates access to multiple bits on the specified port. For example, the hexidecimal number $0900 = 0000 1001 0000 0000 accesses port bits 0 and 3. The mask 0xFF00 selects all of the bits in the port. If a port bit mask is used, the low byte must always be 0.

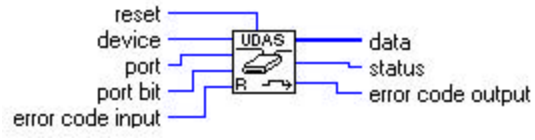



**FIGURE 6.5 uDIOBITRead Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Digital Input/Output Bit Functions* item and select **uDIOBITRead**.

## 6.3.3 uDIOBITWrite

This digital input/output bit function, shown in FIGURE 6.6, *uDIOBITWrite Function*, sets the value(s) of a digital output port bit or bits. This function is also configured for one of the trigger input modes. Refer to section 6.3.2, *uDIOBITRead* for information on port bit input.

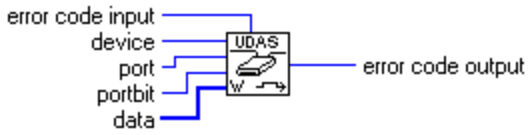



**FIGURE 6.6 uDIOBITWrite Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Digital Input/Output Bit Functions* item and select **uDIOBITWrite**.

# Chapter 7 Support Library: Hardware Counter Functions

## 7.1 Introduction

To configure a 16-bit hardware counter on a UDAS unit for a specified counter mode, use the **uCTRConfigure** function. Each UDAS unit has one 16-bit hardware counter. Use thiscounter for event counting, frequency, period, and pulse measurement. Refer to the UDAS Support Library online manual for detailed information on this function.

To start or stop an event counter, or to disable a frequency or time measurement, use the **uCTREnable** function. When the hardware counter is used as an event counter, **uCTREnable** (with the enable input set to a non-zero value) starts the counting process. If enable input is set to zero, then the counting process stops.

For time measurement modes (frequency, period and pulse measurement), the **uCTREnable** function does not need to be called before the **uCTRRead** function. The **uCTREnable** function with the disable flag halts the measurement in progress, and disables the interrupts. Before it can be reconfigured, the counter must be disabled in any mode.

### Functions

**uCTRConfigure.vi** configures a counter on a UDAS unit for a specified counter mode.

**uCTREnable.vi** starts or stops an event counter, or disables a frequency or time measurement.

**uCTRRead.vi** reads the current count value in event counter mode, or starts a measurement function.

**uCTRStatus.vi** returns the current status of the event-counting or measurement process.

For more information on how to perform an event counting, frequency measurement or time measurement using UDAS 16-bit hardware counter refer to the **UDAS Hardware Counter Sample.vi**.

# 7.2 uCTRConfigure

This function (shown in FIGURE 7.1, *uCTRConfigure Function*) configures a 16-bit hardware counter on a UDAS unit for a specified counter mode.

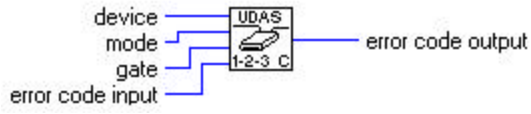



**FIGURE 7.1 uCTRConfigure Function**

## Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Counter Functions* item and select **uCTRConfigure**.

## Counter Modes

### Event Counter

The event counter counts the number of pulses on the input signal from the time the counter is enabled, or is read and reset. Counting occurs when the counter is gated *on* (with the gate input signal) and stops when it is gated *off*.

If the mode selected is Event Counter Mode High, the gate input must be high to enable counting. The LabVIEW numeric value for the event counter with the gate input high is 0. For mode Event Counter Mode Low, the gate input is low to enable counting. The LabVIEW numeric value for the event counter with the gate input low is 4.

The gate input may perform multiple transitions while the counting occurs. The counter may be read at any time and optionally reset. If the counter is not reset, the readings are cumulative; otherwise the result will be counts since the previous reading.

### Period Measurement

In Period Counter Mode, the counter measures the duration of one complete period of the input signal. When the counter is read, the data returned is the period measurement in nanoseconds. Because the counter counts 8 MHz clock cycles, the resolution of this measurement is 125 nanoseconds. The LabVIEW numeric value for the Period Counter Mode with the gate input low is 3.

### Pulse Measurement

The counter measures the duration of the High Pulse or the Low Pulse of the input signal. Since the counter counts 8 MHz clock cycles, the resolution of this measurement is 125 nanoseconds. The LabVIEW numeric value for the high pulse measurement is 1. The LabVIEW numeric value for the low pulse measurement is 9.

### Frequency Measurement

In the Counter Frequency Mode, a gate value (ignored in other modes) must be specified. The counter measures frequency by counting 8 MHz pulses for the duration specified by the gate value, using an internal gate. The frequency measured is returned in Hz (cycles per second). The LabVIEW numeric value for the frequency measurement is 2.

The Gate Value determines the precision of the measurement. Valid gate values are: 1 ms (LabVIEW numeric value is 10), 10 ms (LabVIEW numeric value is 100), 100 ms (LabVIEW numeric value is 1000), and 1 s (LabVIEW numeric value is 10000).

## 7.3 uCTREnable

This counter function (shown in FIGURE 7.2, *uCTREnable Function*) is used to start or stop an event counter, or to disable a frequency or time measurement. Apply any non-zero value to *enable* input to start event counting. Use a value of zero to stop event counting, or to disable a measurement process.

**FIGURE 7.2 uCTREnable Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Counter Functions* item and select **uCTREnable**.

# 7.4 uCTRRead

This counter function (shown in FIGURE 7.3, *uCTRRead Function*) reads the current count value in event-counter modes. For measurement modes, this call can start the measurement or return the result from a completed measurement. On *data* output, frequency readings are returned in Hertz, event counter readings are returned in counts, and period and pulse width measurements are returned in nanoseconds.

In event counter mode, the *reset* input causes a software reset to be performed on a reference count, without missing any counts during reset. In other modes, the reset input will cause a new measurement to be immediately triggered if the current one has been completed.

**FIGURE 7.3 uCTRRead Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Counter Functions* item and select **uCTRRead**.

# 7.5 uCTRStatus

This counter function (shown in FIGURE 7.4, *uCTRStatus Function*) returns the current status of the event-counting or measurement process.

The following status bit codes may be returned:

0x0100 indicates that measurement is in progress (period, pulse, or frequency measurement).

0x0200 indicates that the counter is running.

0x0400 indicates that the counter is not configured.

If the returned status indicates that a measurement is in progress, the data value is not valid. Additional calls to **uCTRRead** or **uCTRStatus** must be made until the measurement is completed.

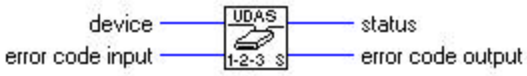



**FIGURE 7.4 uCTRStatus Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the Counter Functions item and select **uCTRStatus**.

# Chapter 8 Support Library: Thermocouple Functions

## 8.1 Thermocouple Functions

### Converting Counts to Temperature

The **uCountsToTemperature** function converts and linearizes previously acquired thermocouple counts values to temperature values. This function uses the counts data for the thermocouples and cold-junction compensation (CJC) readings (such as data that is acquired through **uAIRead** calls). It also uses the gain and input range that the readings were obtained with, and the thermocouple type (type J, K, or T) in order to calculate the thermocouple junction temperature in degrees Celsius.

### Available Signal Termination Panels with Built-in CJC Sensor Channels

To simplify the connections and measurements, various signal termination panels with built-in CJC sensor channels are available from Intelligent Instrumentation. For additional information about this function, see the UDAS Support online manual.

In most instances, thermocouple measurement requires acquiring data from two analog input channels, one to obtain the thermocouple's output voltage and the other to acquire CJC data. If the thermocouple data was acquired by using a signal conditioning module, such as a PCI-5B37 isolated thermocouple block mounted in a 5B termination panel, CJC is already in the measurement and does not need to be supplied separately. In addition to standard thermocouple data conversion, **uCountsToTemperature** supports conversion and linearization of data acquired from PCI-5B37J, PCI-5B37K, and PCI-5B37T non-linearized thermocouple signal conditioning blocks.

## 8.1.1 uCountsToTemperature

This thermocouple function (shown in FIGURE 8.1, *uCountsToTemperature Function*) converts and linearizes acquired thermocouple counts to temperature values.

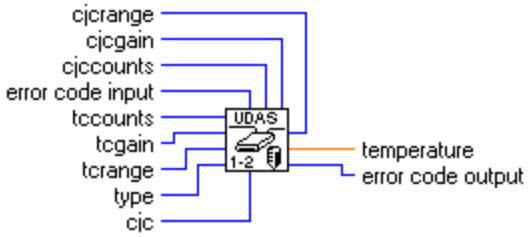



**FIGURE 8.1 uCountsToTemperature Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Utilities Functions* item and select **uCountsToTemperature**.

## 8.1.2 uCountsToTemperatureA

This thermocouple function (shown in FIGURE 8.2, *uCountsToTemperatureA Function*) converts and linearizes previously acquired arrays of the thermocouple and CJC count values to arrays of temperature values.

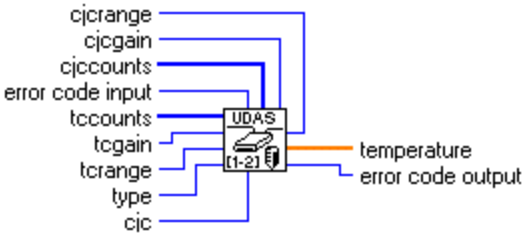



**FIGURE 8.2 uCountsToTemperatureA Function**

### Accessing the Help Page for this Function

Access the online Help manual as described in Chapter 3, *UDAS Support Library: Mandatory & Optional Functions*. In the Contents dialog, double-click the *Utilities Functions* item and select **uCountsToTemperature**. Scroll down the page to locate the **uCountsToTemperatureA** portion of the page.

# Appendix A Error Codes

## A.1 Introduction

All UDAS Support VIs report a 32-bit integer error code at an error code output terminal. This code indicates the success or failure of the function. This appendix contains a complete listing of all error codes.

### A.1.1 Explanation of Error Code Values

**Zero Value:** A zero value is output if the function execution successfully completes without any errors.

**Positive Value:** A positive value returns if the execution function does not complete because of a critical error.

Upper-case C/C++ error code predefined constants have no direct application in LabVIEW. They are provided exclusively for use with Intelligent Instrumentation's U-Link software. A programmer with previous experience in the development of UDAS applications in C/C++ will recognize some of those error definitions and use corresponding numbers in LabVIEW.

## A.2 List of Error Codes

The following list of error codes are located in the UDAS Support Library online manual which can be accessed through the Windows Start menu. Click the UDAS Support Online Manual shortcut and the *Help Topics: Help for LabVIEW UDAS Support Library* dialog opens. The *Contents* tab is the active dialog displayed which contains a list of items beginning with the contents, followed by the different types of functions, and the error codes.

**Successful Code**

0 NO_USB_ERROR

Indicates the UDAS support function has been properly executed.

**Error Codes returned by the UDAS Driver DLL**

1 INVALID_DEVICE_ERR

An invalid device handle was passed by the low-level VI to the UDAS drivers.

2 INVALID_PIPE_ERR

Indicates that an internal error occurred in the UDAS drivers. Contact the factory.

3 OUT_OF_DEVICES_ERR

Too many devices have been opened. The maximum number of eight devices can be open at the same time.

4 DEVICE_OPEN_ERR

The UDAS unit could not be opened. This may occur if another application, such as another LabVIEW VI or SYSCHECK, is currently communicating with the UDAS device.

5 MAILBOX_TEST_ERR

An internal error occurred, indicating it may be a hardware problem. Contact the factory.

6 FIFO_TEST_ERR

An internal error occurred, indicating it may be a hardware problem. Contact the factory.

7 INVALID_PTR_ERR

Indicates that a LabVIEW low-level VI passed an invalid value to the UDAS drivers.

8 SERIAL_NO_ERR

An internal error occurred and the UDAS unit responded improperly to a message. Contact the factory.

9 HOST_BUFFER_ERR

Indicates the specified high-speed analog input buffer size is too large to be allocated by the UDAS drivers. Modify the settings of the **uAIHSEnable** call to decrease the buffer size.

**Error Codes returned by the UDAS unit**

10 EP1_LENGTH_ERR

An internal error has occurred. The computer sent the unit a packet of data having an incorrect length. Contact the factory.

11 EP2_LENGTH_ERR

An internal error has occurred. The computer received a packet of data from the UDAS unit that was the correct length. Contact the factory.

12 EP3_LENGTH_ERR

An internal error has occurred. The computer sent the unit a packet of data having an incorrect length. Contact the factory.

13 EP4_LENGTH_ERR

An internal error has occurred. The computer received a packet of data from the UDAS unit having an incorrect length. Contact the factory.

18 SEQUENCE_ERR

A packet expected as a part of a message to the UDAS device was lost. This error should not occur. Contact the factory.

19 EP3_SEQ_ERR

A packet expected as a part of a message to the UDAS device was lost. This error should not occur. Contact the factory.

20 NO_COMMAND_ERR

The UDAS unit received an invalid command code. This could be caused by using a later version of the UDAS drivers with a UDAS unit which does not support one or more of the advanced driver functions.

21 EE_ADDRESS_ERR

Indicates an internal error in EEPROM addressing. Contact the factory.

22 BUFFER_ERR

Indicates that buffers for high-speed analog input or output data are not available. Decrease the rate of data acquisition for the analog input process, or decrease the number of data points output in the analog output process.

**Multi-function Error Codes**

23 AIMAX_CONFLICT_ERR

Indicates that an attempt was made to configure digital I/O port 1 for input while AI-MAX was in use for analog input channel expansion. Refer to the hardware manual for restrictions on I/O configurations when AI-MAX is in use.

24 T2_CONFLICT_ERR

An internal timer on the UDAS unit is used for analog input and counter frequency measurement. This error indicates that an attempt has been made to simultaneously perform both types of processes.

**Digital I/O Error Codes**

25 DIO_PORT_ERR

The specified digital port number is invalid.

26 DIO_MODE_ERR

The digital I/O mode specified for the port is not supported by this port. This error can be generated if a digital input bit mode is set for a port that does not support bits configured for that mode. Reference the hardware manual for mode restrictions.

27 DIO_NOT_CONFIGURED

Indicates that an attempt was made to enable, write to, or read from a port that has not been configured. Before using a port, verify that it has been configured.

28 DIOB_BIT_ERR

The specified digital I/O bit number is not valid. Valid bit numbers are 0-7.

29 DIOB_MODE_ERR

The specified digital I/O bit mode is not valid for the selected port or bit. This could be caused by attempting to configure one bit in an output port for an input mode, a bit in an input port for an output mode, or a bit in a port for a mode that is not supported by bits in that port. To change port direction, reconfigure the entire port and then configure each bit. Refer to the hardware manual for restrictions on mode assignments.

30 DIOB_ENABLE_ERR

Indicates that an attempt was made to reconfigure a digital I/O bit while the bit was enabled. The bit must be disabled before it can be reconfigured.

**Analog Input Error Codes**

31 AI_PANEL_ERR

The AI-MAX panel selected is not valid (valid range is 0-15), or AI-MAX has not been enabled.

32 CONFIGURE_SEQ_ERR

AI-MAX was enabled on the UDAS unit either after the digital output port used to control AI-MAX channel selection had been configured for an incompatible mode, or analog input was configured prior to enabling AI-MAX. Refer to the hardware manual for information about configuration restrictions on digital I/O ports when AI-MAX is in use. If AI-MAX is not is use, verify that a panel other than 0 has not been selected.

33 AI_DIFF_CONFLICT

An attempt was made to read either the main signal, or the return signal half of a differential channel as a single-ended channel. This is not supported.

34 AI_GAIN_ERR

An invalid or inconsistent gain value has been specified.

35 AI_RANGE_ERR

An invalid or inconsistent range value has been specified.

36 AI_NOT_CONFIGURED

The specified analog input channel was read prior to its configuration.

37 AI_CHANNEL_ERR

An invalid or inconsistent channel number was specified. Verify the channel number is correct, given the single-ended/differential configuration.

38 AI_TIMEOUT_ERR

Indicates that a time-out occurred while waiting for the an analog input conversion to be completed. Contact the factory.

39 AI_DATASIZE_ERR

The number of channels and AI-MAX panels specified is too large to acquire the desired number of samples from each channel. Decrease the number of samples or the number of channels.

**Analog Output Error Codes**

40 AO_CHANNEL_ERR

Indicates that an invalid analog output channel number was specified. This error occurs when an attempt is made to use analog output on a UDAS unit that does not have analog output channels.

41 AO_RANGE_ERR

An invalid range parameter was passed for configuration of an analog output channel.

42 AO_COUNT_ERR

An invalid number of channels were specified in the configuration list for the analog output circuit. Attempting to use analog output on a UDAS unit that does not have analog output channels may result in this error.

**Rate Generator Error Codes**

43 RG_CHANNEL_ERR

The rate generator channel number is invalid.

44 RG_COUNT_ERR

An invalid frequency was specified for the rate generator channel. Refer to the hardware manual for information on minimum and maximum rates that are supported.

45 RG_NOT_CONFIGURED

The rate generator was enabled prior to its configuration.

**Trigger Error Codes**

46 TRIG_MODE_ERR

The specified analog trigger mode is invalid.

### High-speed Analog Input Error Codes

47 AIHS_RUNNING_ERR

This error occurred when an attempt was made to configure a high-speed input process while another high-speed input process was already running on the UDAS unit.

48 AIHS_LIST_ERR

An invalid channel sequence was specified. The channels in the list must range from channel 0 to N, where N depends on the input configuration, either single-ended or differential.

49 AIHS_PARAM_ERR

Indicates the analog input configuration list is incorrectly formatted.

50 AIHS_ZCHAN_ERR

The auto-zero function is not permitted for high-speed analog input.

51 AIHS_START_ERR

An invalid start condition for analog input acquisition was specified.

52 AIHS_STOP_ERR

An invalid stop condition for analog input acquisition was specified.

53 AIHS_PACER_ERR

An invalid pacer source was specified for the high-speed process.

54 AIHS_TRIGGER_ERR

An invalid trigger source was specified for the high-speed process.

55 AIHS_NOT_CONFIGURED

An attempt was made to start a high-speed analog input acquisition before a process was configured.

56 AIHS_NOT_ENABLED

An attempt was made to read data from a high-speed analog input acquisition process, but the process had not been started.

57 TRIG_NOT_CONFIGURED

An attempt was made to enable a high-speed analog input process that depends on a trigger, but the trigger source has not been configured.

58 TRIG_CHAN_ERR

The analog trigger channel selected for the high-speed analog input process is not included among the channels being read by the process. The channel must be one of those being configured for the process to be used as the trigger channel.

59 PAC_NOT_CONFIGURED

The pacer signal for a high-speed analog output process has not been configured.

60 AIHS_CLUSTER_ERR

The cluster value passed to the high-speed analog input read call is invalid.

**High-speed Analog Output Error Codes**

61 AOHS_PACER_ERR

The pacer type specified for the high-speed analog output process is invalid.

62 AOHS_SYNCH_ERR

An invalid trigger source for analog output was selected.

63 AOHS_LIST_ERR

The channels specified for the high-speed analog output process are given in an incorrect order, or an invalid channel number was specified. When both channels are to be output, channel 0 and channel 1 must be specified in ascending order in the list.

64 AOHS_DATA_ERR

An attempt was made to start the high-speed analog output process without sending initial data to the UDAS unit.

65 AOHS_INUSE_ERR

An attempt was made to enable the analog output while the output process was in progress.

66 AOHS_WRITE_ERR

An attempt was made to write new data to the analog output process, but the process is configured in repeat mode, requiring the same data to be output continuously. To change the data in repeat mode, stop the output process and reconfigure the process.

67 AOHS_RECV_ERR

The next data buffer for high-speed analog output was not received quickly enough to maintain continuous output.

68 AOHS_NOT_RUNNING_ERR

An attempt was made to send new data for a high-speed analog output process that was either was not started, or had stopped.

69 AOHS_CONF_ERR

An attempt was made to start high-speed analog output, except the process has not been configured.

**Counter Error Codes**

70 CTR_INVALID_MODE

The counter mode specified in the counter configuration is invalid.

71 CTR_NOT_CONFIGURED

A read or measure operation was attempted on a counter channel that has not been configured.

72 CTR_READ_PROG_ERR

A counter measurement was still in progress when an attempt was made to read the counter data value.

73 CTR_ENABLE_ERR

An attempt was made to reconfigure a counter channel that was running at the time. Disable the counter and then reconfigure the counter.

74 CTR_GATE_ERR

The gate time specified when configuring a counter channel for frequency measurement mode was invalid.

75 CTR_LATCH_ERR

Attempts to read the counter channel failed. This may be caused by attempting to use an input signal with a frequency that is too high.

76 CTR_CHANNEL_ERR

An invalid counter channel number was specified.

**Thermocouple Error Codes**

77 TC_BUFFER_ERR

Unable to allocate memory for thermocouple tables.

78 TC_MEMORY_ERR

Internal memory error in thermocouple table processing. Contact the factory.

79 TC_FILE_OPEN_ERR

The attempt to open a thermocouple table file failed. Verify the FILES=setting is large enough, the table is present in the same directory as the VI, and the file has not been corrupted.

80 TC_TABLE_READ_ERR

The thermocouple file could not be read. The file may be in an incorrect format either because it has been corrupted, or it is not a U-Link .TC file.

81 TC_NO_DATA_ERR

Unable to locate the thermocouple files. Ensure the U-Link .TC files are installed in the same directory as the U-Link DLL.

82 TC_NO_TABLE_ERR

The indicated thermocouple type is either not supported by this version of the driver, or the thermocouple file corresponding to the thermocouple type specified is missing.

83 TC_5B_CJC_ERR

When using 5B37 thermocouple blocks, do not use CJC, as the cold-junction compensation is internally performed in each block.

84 TC_CJC_RANGE_ERR

The cold-junction compensation temperature value is out of range. Verify the termination panel is configured correctly for thermocouple measurements.

85 TC_VOLTAGE_RANGE_ERR

The thermocouple temperature is out of range. The thermocouple may be improperly connected or broken.

**Miscellaneous Error Codes**

86 OPTION_ERR

An invalid option value was specified to the **uSetOption** call.

87 NO_AO_ERR

An analog output operation was attempted on a unit which does not have analog output support.

88 IO_NOT_READY_ERR

An attempt to communicate with the UDAS device occurred before the unit was fully operational after a computer reboot, or after being connected to the USB port.

89 AI_RESET_ERR

During a reset operation, an internal error occurred on the UDAS unit. Contact the factory.

90 EP3_DATA_ERR

There was an internal error in USB communication with the UDAS unit. Contact the factory.

91 OVERLOAD_ERR

Indicates that interrupts from digital input bits configured for change-of-state modes are occurring too rapidly. The maximum aggregate input rate cannot exceed 2kHz for more than 20ms. Check the source of input pulses to the digital input bits. You may need to debounce the output of a switch, if the output pulses at a high frequency for long period of time when changing states.

92 BURNIN_ERR

This internal error indicates that there may be a problem with the hardware. Contact the factory.

93 AIHS_DELAY_ERR

The trigger delay size is too large. When a high-speed analog input process is configured for Stop On Trigger mode, the total number of trigger delay counts may not exceed 32767. Note: To determine the total number of counts, multiply the number of delay counts per channel (set by the AIHSEnable function call) by the number of channels configured for a high-speed data acquisition process.

# Index


**A**

A/D conversions
  high-speed analog input 4-8
  initiating for high-speed analog input 4-12
AI-MAX 4-4
  256 channel expansion 4-1
  termination panels 4-5
analog input
  channel
    setting differential mode 4-3
    setting gain 4-3
    setting input range 4-3
    setting single-ended mode 4-3
    setting zero-reference flag 4-3
    termination panel 4-3
  channels, configuring a list of 4-3
  configuring list of channels 4-10
  data, format of 4-14
  functions 4-1
  high-speed, pacer 4-6
  high-speed, starting, stopping process 4-13
analog output
  functions 5-1
  rate generator channel, configuring 5-4
  waveform, downloading a segment of 5-8
analog trigger circuit 4-7
application
  creating with UDAS support library 3-1
  VIs, sample 1-2
applications directory, sample 2-6
array
  count elements 4-3, 6-4
  data 4-4, 5-2
arrays
  previously acquired of thermocouple & CJC
    count values 8-2
  temperature values 8-2

**B**

bit
  codes, status 7-5
  functions
    digital input, output 6-4
  writing to, configured for output modes 6-5
buffer
  information, high-speed analog input 4-15
  information, high-speed analog input
    process 4-13

**C**

channel
  list specified by VI inputs 4-3, 5-1
channels
  count input
    number to configure 4-3
circular buffer 4-11
CJC sensor channels, built-in 8-1
clock
  TTL-level source, external pacer 4-12
clustersize output
  high-speed analog input 4-10
common format data
  converting to floating point voltage values 4-14
configuration requirements, system 2-1
configuring
  list of analog output channels 5-1
converting
  common format data to floating point voltage
    values 4-14
  linearizing acquired thermocouple counts
    values to temp values 8-1
counter
  event, frequency, period, and pulse
    measurement 7-1
  frequency mode 7-3
  incrementing signal edges 6-5
  measuring a complete period 7-2
  modes 7-2

**D**

data
  output 7-4
  reading, acqired in high-speed analog input
    process 4-14
  returning, acquired in high-speed analog input
    process 4-13
  values 5-2
data acquisition

high-speed analog input
start on command 4-11
start on trigger 4-11
stop on command, 4-11
stop on terminal count 4-11
stop on trigger with delay 4-11
port level 6-1
specifying start and stop modes 4-10
data array 5-2
definitions
uDIOConfigureList 6-1, 6-4
uDIORead 6-1, 6-4
uDIOWrite 6-1, 6-4
deinitializing U-Link drivers 3-11
diagram
wiring connetcion information 3-1
digital
input, output bit function, configuring 6-4
input, output bit function, general purpose 6-6
input, output port functions 6-1
input, output port functions, reading 6-2
input, output port, writing data to 6-3
output port bit or bits 6-7
output port bits
setting the values for 6-4
directory
LabVIEW 2-3
samples applications 2-6
UDAS support library 2-3

**E**

error code
input and output terminals 3-7
error codes 1-3
explaination of values A-1
list A-1
analog input error codes A-4
analog output error codes A-5
counter error codes A-8
digital I/O error codes A-3
error codes returned by UDAS unit A-2
high-speed analog input error codes A-6
high-speed analog output error codes A-7
miscellaneous error codes A-9
multi-function error codes A-3
rate generator errors A-5
returned by UDAS driver DLL A-1
thermocouple error codes A-8
trigger error codes A-5
values
positive value A-1
zero value A-1
error, critical 3-7
event counter 7-2
disabling frequency or time measurement 7-1
gate input high 7-2
gate input low 7-2
reset input 7-4
starting, stopping 7-1, 7-3
event counting
returning current status 7-5
examples
LabVIEW, UDAS 2-6
labviewUDAS
analog input sample.vi 2-6
analog output sample.vi 2-6
digital input bit sample.vi 2-6
digital IO sample.vi 2-6
digital output bit sample.vi 2-6
high-speed analog input sample.vi 2-7
thermocouple sample.vi 2-7
typej.tc 2-7
typej5b.tc 2-7
typek5b.tc 2-7
typet.tc 2-7
typet5b.tc 2-7

**F**

floating point voltage values 5-2
converting common format data to 4-14
frequency measurement 7-3
function
block &VI file listing 2-2
VI icons 2-8
wiring connection information 3-1
functions
high-speed
analog input 4-7
analog input data acquisition, hardware controlled 4-1

high-speed analog input, required for 4-6
high-speed analog output 5-3
mandatory 3-2
initialization 3-2
mandatory used in diagrams 3-1
optional 3-3
software controlled analog input data acquisition 4-1
trigger 4-6

**G**

gain and input range 8-1
gate
input 7-2
value 7-3
values, frequency measurement 7-3

**H**

help
pages, online, for functions 3-1
using online 1-2
high pulse 7-3
high-speed
analog output
obtaining status information for 5-7
trigger functions 5-4
analog output processes
enabling or disabling 5-6
data acquisition
required functions 5-3
data acquisiton
required functions 4-6
start and stop modes 4-10
high-to-low signal edges
incrementing a counter 6-5

**I**

icons
function VI 2-8
incrementing a counter 6-5
input signal
counting pulses 7-2
installation
software 2-2

**L**

labview directory 2-3
library
data acquisition I/O and control functions 1-2
support 1-2
license agreement, software 2-1
low pulse 7-3
low-level LabVIEW VIs 1-2

**M**

measurement
frequency 7-3
period 7-2
process, disabling 7-3
pulse 7-3
resolution of 7-3
modes
counter 7-2
high-speed analolg input 4-11
stop on command 4-11
monitoring
high-speed analog input process 4-15

**O**

online help, using 1-2

**P**

pacer 4-10
configuring analog input channels 4-7
high-speed analog input 4-6
input, high-speed analog input 4-12
using external source 4-12
pacers and triggers
start and stop modes 4-10
pacing signal
high-speed analog input 4-12
panel control 4-2
period
measurement 7-2
period counter mode 7-2
port
bit
digital input, output, configuring 6-4
read or written 6-6

functions
digital input, output 6-1
list
specifying 6-1, 6-4
port bit
input, integer value 6-6
ports
dimensioning number to read 6-2
previously configured AOchannel, writing data to 5-2
pulse
measurement 7-3
width measurements 7-4

**R**

rate generator
calculating output frequency 4-8
configuring analog input channel 4-8
enabling, disabling a channel 4-9
functions 4-6
high-speed analog input 4-7
using on-board analog input 4-12
readings, counter 7-2

**S**

sample
applications directory 2-6
driver-based program 3-5
sampling
one or multiple inputs with time-based source, triggering 4-1
signal
edges, incrementing 6-5
software
installation 2-2
license agreement 2-1
start mode 4-11
status
bit codes 7-5
information, enabling high-speed process 4-7
information, high-speed analog input 4-15
stop mode 4-11
support
functions, UDAS 2-4
library 1-2
files 2-2
function calls 1-2
UDAS online reference manuals 2-3
system configuration requirements 2-1

**T**

termination panels
AI-MAX 256 channel AI 4-1
maximum number 4-2
number installed in AI-MAXsystem 4-5
selecting a rate generator for 3-3
thermocouple
calculating thermocouple junction temperature 8-1
function 8-2
time measurement modes 7-1
trigger 4-10
event
high-speed analog input process 4-6
function,high-speed analog input 4-7
signal, generating 4-7
sources, configuring 4-7
TTL-level source clock 4-12

**U**

UDAS
analog trigger circuit 4-7
board identification number 3-3
configuring specified counter mode 7-1
function example files 3-10
hardware and firmware versions 3-3
incrementing a counter on signal edges 6-5
incrementing a counter, any mode 6-5
internal errors, information 3-3
LabVIEW support library, using 4-1
low-level VIexecution 3-11
low-to-high signal edges
incrementing a counter 6-5
optional parameters, setting 3-3
rate generators
high-speed analog output 3-3
high-speed data acquisition 3-3
serial number 3-3

support functions 2-4
support library directory 2-3
support online reference manuals 2-3
support VI 1-2
UDAS function example files 3-11

**V**

voltage values, floating point 5-2

**W**

waveform
continuous analog output 5-8
wiring connection information, function 3-1
writing
bit configured for output mode 6-5
data to analog output channels 5-1